FlowViewer

FlowViewer
FlowGrapher
FlowTracker

Version 3.3

User's Guide

1. Introduction

FlowViewer is an open source set of tools that provide a convenient web-based user interface to Mark Fullmer's flow-tools suite. The tools provide additional graphing and tracking features by utilizing open source software including Lincoln Stein's GD, Martien Verbruggen's GD::Graph, and Tobias Oetiker's RRDtool packages.

The umbrella FlowViewer package consists of FlowViewer, FlowGrapher, and FlowTracker. Each of these tools uses a web interface to collect filtering information and applies the filter to netflow data captured and stored by flow-tools, resident on the same host. The processing of each of the tools is configured via a common configuration file. Guidance for using each of these tools is presented in separate sections below.

2. Installation

Quick Instructions for an Upgrade

1. Untar the package into new cgi-bin subdirectory
2. Configure FlowViewer_Configuration.pm variables as necessary
3. Replace old logos with new logos (might be done automatically)
4. Configure FlowViewer_Configuration.pm to point to existing FlowTracker_Filter and FlowTracker_RRDtool directories
5. Stop old Flowtracker_Collector and FlowTracker_Grapher
6. Start new Flowtracker_Collector and FlowTracker_Grapher
7. Modify NamedInterfaces_Devices file if using them
8. Use included 'User Relay' scripts if desired (see below)

Quick Instructions for Installation:

1. Untar into cgi-bin subdirectory

For FlowViewer

2. Configure FlowViewer_Configuration.pm variables as necessary
3. Point browser to FlowViewer.cgi

For FlowGrapher

4. Install GD, GD::Graph
5. Configure FlowViewer_Configuration.pm variables as necessary
6. Point browser to FlowGrapher.cgi

For FlowTracker

7. Install RRDtool
8. Create FlowTracker_Filter and FlowTracker_RRDtool directories
9. Configure FlowViewer_Configuration.pm variables as necessary
10. Start FlowTracker_Collector, FlowTracker_Grapher in background

11. Point browser to FlowTracker.cgi

More detailed information:

1. Getting and untarring the package:

   Obtain the latest version of FlowViewer from the FlowViewer website:

   http://ensight.eos.nasa.gov/FlowViewer/

   A good way to get a copy is to use the ‘wget’ program. For example:

   ```
   /htp/cgi-bin >wget http://ensight.eos.nasa.gov/FlowViewer/FlowViewer_3.3.tar
   ```

   From your system’s cgi-bin directory:

   ```
   /htp/cgi-bin/ 103 >tar -xvf FlowViewer_3.3.tar
   FlowViewer_3.3/
   FlowViewer_3.3/FlowViewer_Configuration.pm
   FlowViewer_3.3/flowcap
   FlowViewer_3.3/FlowGrapher.cgi
   FlowViewer_3.3/FlowGrapher_Colors
   FlowViewer_3.3/FlowGrapher_Main.cgi
   FlowViewer_3.3/FlowGrapher.png
   FlowViewer_3.3/FlowGrapher_Relay.cgi
   FlowViewer_3.3/FlowGrapher_Save.png
   FlowViewer_3.3/FlowGrapher_Sort.cgi
   FlowViewer_3.3/FlowTracker.cgi
   FlowViewer_3.3/FlowTracker_Collector
   FlowViewer_3.3/FlowTracker_Dumper.cgi
   FlowViewer_3.3/FlowTracker_Grapher
   FlowViewer_3.3/FlowTracker_Group.cgi
   FlowViewer_3.3/FlowTracker_Main.cgi
   FlowViewer_3.3/FlowTracker.png
   FlowViewer_3.3/FlowTracker_Relay.cgi
   FlowViewer_3.3/FlowViewer.cgi
   FlowViewer_3.3/FlowViewer_CleanASCache
   FlowViewer_3.3/FlowViewer_CleanFiles
   FlowViewer_3.3/FlowViewer_CleanHostCache
   FlowViewer_3.3/FlowViewer_Main.cgi
   FlowViewer_3.3/FlowViewer.png
   FlowViewer_3.3/FlowViewer_Relay.cgi
   FlowViewer_3.3/FlowViewer_Save.cgi
   FlowViewer_3.3/FlowViewer_Save.png
   FlowViewer_3.3/FlowViewer_Utilities.pm
   FlowViewer_3.3/Generic_Logo.jpg
   FlowViewer_3.3/NamedInterfaces_Devices
   FlowViewer_3.3/NamedInterfaces_Exporters
   FlowViewer_3.3/README
   ```

This has created a cgi-bin subdirectory called FlowViewer_3.3 which includes the whole package. It may be the case that you have created this directory as a user that is not the same as the owner of the web server process. The web server may, depending on your configuration (more later) need to write into this directory. If that is the case, you must give this directory adequate ‘write’ permissions. I generally provide it with ‘0777’ (e.g., chmod 0777 /htp/cgi-bin/FlowViewer_3.3) since my web server process owner is ‘apache.’

If you plan to use FlowTracker, you’ll need to create directories to hold the permanent filter files and rrdtool databases that will be created. These are defined by the $filter_directory and $rrdtool_directory parameters. If you’ve been using an earlier version of FlowViewer, and you’ve been using the FlowTracker tool, you’ll want to either set the $filter_directory and $rrdtool_directory parameters to the existing directories, or create new directories and move any existing FlowTracker filters and rrdtool databases to there.

2. Latest Release Information

```
#
# Version 3.3 Release Notes
#
# New Capabilities
#
#   1. Some devices will now have 'named interfaces' (thanks C. Kishimoto)
#   2. The user can now save filters of interest and recall them later
#   3. Data can now be analyzed by Exporter ID (in addition to device name)
#   4. Users can now set thresholds on FlowTrackings, and be alerted
#   5. FlowViewer now provides Pie Charts
#   6. Capability added to apply a Sampling Multiplier to output
#   7. FlowTrackings now have a '3 year' graph
#   8. The user can now generate text listings of FlowTracker output
#   9. Filtering on next-hop has been added
#  10. Logging has been made more flexible (e.g., less data)
#  11. Preserve latest three notations (was keeping first three)
#  12. Can now specify and display time-zones
#  13. A hook has been provided for a User Logo with link out of FlowViewer
#  14. New file cleanup scripts have been added
#  15. Unit Conversion capability has been added (thanks C. Kishimoto)
#  16. Can now graph Flows, Packets as well as Octets (thanks E. Lautenschlaeger)
#  17. Improved AS name resolution (thanks S. Cardus)
#
# New Scripts and Files:
#
#  FlowGrapher_Sort.cgi        Sorts FlowGrapher Detail Lines by column
#  FlowTracker_Dumper.cgi      Invoked by link in Trackings, prints text values
#  FlowViewer_Save.png         New logo with links for saving filters, reports
#  Flowgrapher_Save.png        New logo with links for saving filters, reports
#  FlowViewer_CleanASCache     Tool used to remove obsolete AS name resolutions
#  FlowViewer_CleanFiles       Tool used to remove old intermediate files
#  FlowViewer_CleanHostCache   Tool used to remove obsolete host name resolutions
#  FlowViewer_Relay.cgi        Optional for pointing users to new version (see Notes)
#  FlowGrapher_Relay.cgi       Optional for pointing users to new version (see Notes)
#  FlowTracker_Relay.cgi       Optional for pointing users to new version (see Notes)
#  flowcap                     Optional start-up script for flow-tools and FlowTracker
#
#  NamedInterfaces_Devices     Holds interface names for SNMP indexes for devices
#  NamedInterfaces_Exporters   Holds interface names for SNMP indexes for exporters
#  FlowViewer_SavedFilters     Created during processing to hold saved filters
#
# Notes:
#
#  Many thanks to Carles Kishimoto, Eric Lautenschlaeger, and Sean Cardus for
#  their ideas and code contributions. Thanks to Dario La Guardia for pointing out
#  a graphing problem that turned out to be a rounding error in FlowGrapher. Credit
#  to Peter Hoffswell for the idea of linking the tools.
#
#  There are no new software dependencies with FlowViewer version 3.3, however
#  Named Interfaces now requires Javascript in the browser to operate.
#
#
#  Using the 'Relay' scripts (these are optional)
#
#   If you have other users and you would like to point them to the new version,
#   copy the included 'Relay' scripts over the old FlowViewer.cgi, FlowGrapher.cgi,
#   and FlowTracker.cgi scripts in the last version's directory.
#
#   For example:
#
#    In the directory /htp/cgi-bin/FlowViewer_3.2:
#
#    mv FlowViewer_Relay.cgi  FlowViewer.cgi
#    mv FlowGrapher_Relay.cgi FlowGrapher.cgi
#    mv FlowTracker_Relay.cgi FlowTracker.cgi
#
#   Then, when the user goes to the old FlowViewer, he will be provided a link to
#   the new FlowViewer, and asked to change his bookmarks.
#
#
#  Setting up crontab file for cleaning FlowViewer file:
#
#   min   hr  dom  moy  dow    command
#
#     5   0    *    *    *     /htp/cgi-bin/FlowViewer_3.3/FlowViewer_CleanFiles
#                                > /htp/cgi-bin/FlowViewer_3.3/cleanup.log
#                                2 >> /htp/cgi-bin/FlowViewer_3.3/cleanup.log
#
#  The file cleanup is controlled by parameters in FlowViewer_Configuration.pm:
#
#   $remove_workfiles_time   = 86400;
#   $remove_graphfiles_time  = 7*86400;
#   $remove_reportfiles_time = 7*86400;
```

3. Dependencies

FlowViewer requires that you have flow-tools, flow data files, a web-server, Perl, and the FlowViewer package all installed on the same machine.

You will, of course, need flow-tools. Written by Mark Fullmer, flow-tools versions up to 0.68 are available at:

http://www.splintered.net/sw/flow-tools/

Paul Komkoff Jr., et. al. are keeping a newer ‘fork’ of flow-tools at:

http://code.google.com/p/flow-tools/

If you are planning on using FlowGrapher, you will need to install Lincoln Stein's GD, and Martien Verbruggen's GD::Graph packages. They can be found at:

GD package: *http://search.cpan.org/~lds/GD-2.30/*
GD::Graph package: *http://search.cpan.org/~mverb/GDGraph-1.43/*

If you are planning on using FlowTracker, you will need to install Toby Oetiker’s RRDtool package. This package can be found at:

http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/pub/rrdtool-1.2.13.tar.gz

For each of these you should make sure you have the latest stable versions.

4. Contents of the FlowViewer Distribution

FlowViewer_Configuration.pm

This file contains parameters that configure and control the FlowViewer, FlowGrapher, and FlowTracker environments. This package should remain in the same directory that the CGI scripts are in.

FlowViewer_Utilities.pm

This file contains processing used by multiple programs (e.g., to create the Report Parameters output for each tool, and other utilities (e.g., 'epoch_to_date' which converts between typical date formats and 'seconds since 1972') that are invoked by other scripts. This package should be placed in the same directory as the CGI scripts.

FlowViewer.cgi

This script produces the web page which provides the user the form for entering analysis selection criteria for FlowViewer. Version 3.1 re-organized the processing. FlowViewer.cgi is now the old create_FlowViewer_webpage. This change permits the input date and time to be updated with each invocation.

FlowViewer_Main.cgi

This script responds when the user completes the selection criteria form and submits the 'Generate Report' command. The script creates a flow-tools filter file based on the selection criteria. Based on the input time period, the script concatenates the relevant flow-tools data files for the selected device. The location of the flow-tools raw data files is specified via the 'flow_data_directory' parameter. The script then invokes the selected statistics/print report flow-tools program and reformats the output into HTML. An option is available in FlowViewer_Configuration to have this script use the NDBM capability (for caching resolved host names) instead of the default GDBM capability for users whose Perl distribution does not have GDBM.

FlowViewer.png

The FlowViewer logo. Leave this file in the 'cgi-bin_directory', the FlowViewer.cgi script will place a copy of the image in 'reports_directory'. This logo now has embedded links in it that permit you to easily switch between FlowViewer tools. If you have generated a report or graph and click on an embedded link, it will bring up the requested tool with the existing filter criteria pre-filled.

FlowViewer_Save.png

The FlowViewer_Save logo with links. Leave this file in the 'cgi-bin_directory', the FlowViewer.cgi script will place a copy of the image in 'reports_directory'. This image contains mapped links to the other tools as well as links for saving the filter used or the report generated.

FlowGrapher.cgi

This script produces the web page which provides the user the form for entering analysis selection criteria for FlowGrapher. Version 3.1 reorganized the processing FlowGrapher.cgi is now the old create_FlowGrapher. This change permits the input date and time to be updated with each invocation.

FlowGrapher_Main.cgi

This script responds when the user completes the FlowGrapher selection criteria form and submits the 'Generate Graph' command. The script creates intermediate processing files exactly like FlowViewer above. The script then parses intermediate output, fills time buckets, and generates a graphic image. Textual output accompanies the graph. An option is available in FlowViewer_Configuration to have this script use the NDBM capability (for caching resolved host names) instead of the default GDBM capability for users whose Perl distribution does not have GDBM.

FlowGrapher_Sort.cgi

This script is invoked when the user clicks on a column header for the Detail Lines of a FlowGrapher report. The textual data on the page is sorted and re-presented.

FlowGrapher.png

The FlowGrapher logo. Leave this file in the 'cgi-bin_directory', the FlowGrapher.cgi script will place a copy of the image in 'graphs_directory'. This logo now has embedded links in it that permit you to easily switch between FlowViewer tools. If you have generated a report or graph and click on an embedded link, it will bring up the requested tool with the existing filter criteria pre-filled.

FlowGrapher_Save.png

The FlowGrapher_Save logo with links. Leave this file in the 'cgi-bin_directory', the FlowGrapher.cgi script will place a copy of the image in 'reports_directory'. This image contains mapped links to the other tools as well as links for saving the filter used or the report generated.

FlowGrapher_Colors

This file contains a translation between textual color names and their RGB value counterparts. This file controls colors for both FlowGrapher, and FlowTracker_Grapher. The colors that start with ‘auto’ will enable you to create Groups more easily by automatically selecting the next color from a pre-defined family of colors. The user may add as many colors as desired. If you add colors, you must restart the FlowTracker_Grapher script.

FlowTracker.cgi

This script produces the web page which provides the user the form for entering analysis selection criteria for FlowTracker. The script also provides the user with the ability to review, revise, or remove existing trackings. FlowTracker.cgi was revised for version 3.1.

FlowTracker_Main.cgi

This script responds when the user completes the FlowTracker selection criteria form and submits the 'Establish Tracking' command. The script responds to the users desire to create, remove, or revise a tracking.

FlowTracker_Group.cgi

This script is invoked by FlowTracker_Main.cgi whenever a user wishes to define a tracking group. When this is the case (i.e., user has selected the ‘Group’ pulldown) any filter criteria entered is ignored since it is not required for a group. A group simply points to existing trackings for which filter criteria has already been defined.

FlowTracker_Collector

The script is started once by the user and placed in the 'background'. The script will execute and then sleep for the duration of a five minute period, essentially running every five minutes. For each existing tracking, the script applies the associated filter to the flow data and extracts the amount that occurred during a 5-minute window approximately 30 minutes earlier. This is to permit long-running flows to have been exported and available to the collector. The script then divides the total bits by 300 seconds to get an average bits-per-second rate during the period. The data point is then provided to RRDtool for storage.

FlowTracker_Grapher

The script is started once by the user and placed in the 'background'. The script will execute and then sleep for the duration of a five minute period, essentially running every five minutes. The script runs the RRDtool graph function for each existing tracking. Daily, weekly, monthly, and yearly graphs are updated with the latest information. The script creates an html page for each tracking that includes the filter parameters and the four graphs. The script also creates an overall web page ($tracker_webpage) that provides links to all active tracking pages.

FlowTracker_Dumper.cgi

This script is invoked when the user clicks on a link within the FlowTracking graph labeled '[List values]'. The script dumps the RRDtool contents onto a web page.

FlowTracker.png

The FlowTracker logo. Leave this file in the 'cgi-bin_directory', the FlowTracker.cgi script will place a copy of the image in 'tracker_directory'. This logo now has embedded links in it that permit you to easily switch between FlowViewer tools. If you have generated a report or graph and click on an embedded link, it will bring up the requested tool with the existing filter criteria pre-filled.

FlowViewer_Save.cgi

This script moves temporary save files into a permanent residence as defined by either the 'reports_directory' or 'graphs_directory' environment variables.

FlowViewer_CleanFiles

A utility for cleaning out temporary files that have been left over from debugging (e.g. $debug_files = 'Y'). Files older than the following configurable parameters are removed:

```
  $remove_workfiles_time   = 86400;
  $remove_graphfiles_time  = 7*86400;
  $remove_reportfiles_time = 7*86400
```

See above for *crontab* settings for running this automatically.

FlowViewer_CleanASCache

A utility for cleaning out from the AS resolving cache ($as_file) a resolved AS name that is no longer valid.

FlowViewer_CleanHostCache

A utility for cleaning out from the DNS resolving cache ($names_file) a resolved host name that is no longer valid.

FlowViewer_Relay.cgi, FlowGrapher_Relay.cgi, FlowTracker_Relay.cgi

Short scripts that refer users from version 3.2 to version 3.3. This keeps you from having to notify users to go to a different web site.

flowcap

A generic shell script used for starting up flow-captures and FlowTracker_Collector, and FlowTracker_Grapher. The user must configure this file.

Generic_Logo.jpg

This image is to be replaced by your own image that can point back to anywhere (e.g., your overarching NMS system.)

5. Configuring for your environment

The file FlowViewer_Configuration.pm is used to configure each of FlowViewer, FlowGrapher, and FlowTracker. Most of the parameters in the file do not need to be changed. Those that might require change are discussed below:

Parameter	Description	Example
$ENV{PATH}	Set this variable to include directories to your basic system commands (e.g., rm, mv, etc.)	$ENV{PATH} = ':/usr/local/bin:/usr/sbin';
$FlowViewer_server	This variable should be set to the IP address of the machine that is running your flow-tools, web-server, and the FlowViewer software.	$FlowViewer_server = “192.168.0.1";
$FlowViewer_service	Set this parameter according the service which your web-server is running. The options are ‘http’ or the encrypted ‘https.’	$FlowViewer_service = "https";
$reports_directory	This is the directory into which you will put FlowViewer reports that you wish to save using the ‘save’ option from the report. The directory should be somewhere beneath your …/htdocs directory. IMPORTANT: Must have adequate ‘write’ permissions so that the web-server can write into this directory as necessary.	$reports_directory = "/htp/htdocs/FlowViewer";
$reports_short	This parameter is used within scripts that are being run by the web-server to locate directories off of the web-server default short-cuts. Typically, the web-server omits the directory information pointing to the root of the ‘htdocs’ and ‘cgi-bin’ directories. See this in comparison to the parameter above.	$reports_short = "/FlowViewer";
$graphs_directory	This is the directory into which you will put FlowGrapher output that you wish to save using the ‘save’ option from the report. The directory should be somewhere beneath your …/htdocs directory. IMPORTANT: Must have adequate ‘write’ permissions so that the web-server can write into this directory as necessary.	$graphs_directory = "/htp/htdocs/FlowGrapher";
$graphs_short	This parameter is used within scripts that are being run by the web-server to locate directories off of the web-server default short-cuts. Typically, the web-server omits the directory information pointing to the root of the ‘htdocs’ and ‘cgi-bin’ directories. See this in comparison to the parameter above.	$graphs_short = "/FlowGrapher";
$tracker_directory	This is the directory which will be used to store your Tracking files and graphs. Each tracking will be structured as a subdirectory of this directory, where the subdirectory contains an html page and four RRDtool graphs. IMPORTANT: Must have adequate ‘write’ permissions so that the web-server can write into this directory as necessary. The directory should be somewhere beneath your …/htdocs directory.	$tracker_directory = "/htp/htdocs/FlowTracker";

$tracker_short	This parameter is used within scripts that are being run by the web-server to locate directories off of the web-server default short-cuts. Typically, the web-server omits the directory information pointing to the root of the ‘htdocs’ and ‘cgi-bin’ directories. See this in comparison to the parameter above.	$tracker_short = "/FlowTracker";
$cgi_bin_directory	This is the directory into which you have placed the FlowViewer scripts. It should somewhere beneath you system’s main cgi-bin directory. IMPORTANT: Must have adequate ‘write’ permissions so that the web-server can write into this directory as necessary.	$cgi_bin_directory = "/htp/cgi-bin/FlowViewer_3.3";
$cgi_bin_short	This parameter is used within scripts that are being run by the web-server to locate directories off of the web-server default short-cuts. Typically, the web-server omits the directory information pointing to the root of the ‘htdocs’ and ‘cgi-bin’ directories. See this in comparison to the parameter above.	$cgi_bin_short = "/FlowViewer_3.3";
$work_directory	This directory is used to hold intermediate files generated during processing, including save files created in case someone wants to save the file. IMPORTANT: Must have adequate ‘write’ permissions so that the web-server can write into this directory as necessary. Also, some intermediate files are quite large, so the size of the partition that holds this directory should be of adequate size.	$work_directory = "/tmp”;
$names_directory	This directory specifies where you would like to store the ‘names’ file created in the process of resolving IP addresses to hosts names. The file is used to cache names for much quicker retrieval than using the ‘dig’ function to get them. It is a good idea to keep this file in a more permanent place (e.g., not /tmp) since temporary directories are cleaned out on system reboots, etc.. IMPORTANT: Must have adequate ‘write’ permissions so that the web-server can write into this directory as necessary.	$names_directory = "/htp/cgi-bin/FlowViewer_3.3";
$filter_directory	This directory is used to store permanent filter files associated with the long-term trackings established using FlowTracker. IMPORTANT: Must have adequate ‘write’ permissions so that the web-server can write into this directory as necessary. This directory must be kept around through FlowViewer version updates if the user wishes to continue with the existing trackings.	$filter_directory = "/htp/cgi-bin/FlowViewer_3.3/FlowTracker_Filters";
$rrdtool_directory	This directory is used to store permanent RRDtool files associated with the long-term trackings established using FlowTracker. IMPORTANT: Must have adequate ‘write’ permissions so that the web-server can write into this directory as necessary.	$rrdtool_directory = "/htp/cgi-bin/FlowViewer_3.3/FlowTracker_RRDtool";

	This directory must be kept around through FlowViewer version updates if the user wishes to continue with the existing trackings.	
$flow_data_directory	This is the directory that sits at the top of subdirectories that store raw flow-tools netflow data. Note that if you are using EXPORTER_ID to distinguish your devices, instead of storing each device’s netflow data in a separate directory, then you can ignore this field and use the $exporter_directory.	$flow_data_directory = "/htp/flows";
$exporter_directory	This is the directory that stores all of the netflow data that you are exporting when you are capturing data from more than one device onto the same port. This is opposed to capturing data from different devices on different ports (i.e., multiple instantiations of *flow-capture*), and then storing each device’s netflow data into a different directory, distinguished by *device_name*.	$exporter = “/htp/flows/all_routers”;
$flow_bin_directory	This directory contains all of the flow-tools programs.	$flow_bin_directory = "/usr/bin";
$rrdtool_bin_directory	This directory holds the rrdtool binary.	$rrdtool_bin_directory = "/usr/local/rrdtool-1.2.12/bin";
$actives_webpage	This is the name of the file that will hold the overall list of all trackings providing a single entry point for users to select a tracking of interest. This file will be created and placed into the directory specified by the ‘tracker_directory’ parameter.	$actives_webpage = "index.html";
$trackings_title	This parameter defines the title that will appear on the web page that lists all of the trackings.	$trackings_title = “Your Company Name”;
$user_logo	This defines the name of the image file containing the user’s logo. This option if used will place the user’s logo next to the FlowViewer logos on all pages providing an exit link for the user.	$user_logo = “Generic_Logo.jpg”;
$user_hyperlink	Defines the hyperlink to be associated with the User logo. For example, this could point to the user’s overarching Network Management System.	$user_hyperlink = www.yourcompany.com/NMS;
$version	Simply the current FlowViewer version number, used to differentiate between versions.	$version = “3.3”;
@devices	This array holds a list of all of the different devices you are collecting netflow data from. The FlowViewer can use a flow-tools data directory layout that has a particular device at the top. A typical flow-tools directory looks like: /flows/router_1/2006/2006-07/2006-07-04 The device name (router_1) is obtained from this array. Populate this array with your device names. If your flow-data file structure does not include a	@devices = ("router_1","router_2","router_3");

	device name, for example you are collecting only from one device, set the @devices array to empty (i.e., @devices = ("");) and set: $no_devices_or_exporters = “Y”; Note that version 3.3 introduces the “Exporter” option which allows users to collect all devices on a single port and separate them via EXPORTER_ID. If you are taking the “Exporter” approach exclusively (i.e., you are not also using devices as described here) you may comment out this parameter. See next parameter	
@exporters	If you are collecting from all of your devices onto a single flow-capture port, you may use $exporter[n] to separate the data. If so, uncomment this parameter. Each entry in this array is formed like this: exporter_ip_address : exporter_name On the FlowViewer input screens you will then see a pulldown with each exporter defined by exporter name. Internal searches will be based on the associated IP address.	@exporters = ("192.168.100.1:New York Router","192.168.100.2:Prague Router");
$no_devices_or_exporters	You need to set this parameter to “Y” if you are using neither devices nor exporters, you are simply collecting data (probably from just one device) into one directory. If you have devices and/or exporters, this field should be left at “N”.	$no_devices_or_exporters = “N”;
$flow_capture_interval	This variable defines the length of time beyond your specified end_time up to which the script will continue to parse through the flow_data looking for flows that occurred during your specified time period, but were exported from the router after the time_period. Some long flows, with a lot of data, may not complete and be exported from the router until well after your specified end_time.	$flow_capture_interval = 30 * 60; # Continue to look for flows 30 minutes beyond
$flow_file_length	This parameter defines how long each of your flow data files is. This is set via *the flow-tools flow-capture* command and defaults to 15 minutes.	$flow_file_length = 15 * 60
$start_offset	This parameter specifies how far back before the current time to specify the start_time for your FlowViewer or FlowGrapher run. These are the default start and end times that appear on your filter input screens.	$start_offset = (90 * 60); # e.g., 90 minutes ago
$end_offset	This parameter specifies how far back before the current time to specify the end_time for your FlowViewer or FlowGrapher run. The example below and the one above will specify a one hour period occurring approximately 30 minutes ago.	$end_offset = (30 * 60); # e.g., 30 minutes ago
$use_even_hours	If set to “Y” this parameter will cause the start and end times of your report period to be set on the hour.	$use_even_hours = "Y";

$N	Different organizations store captured netflow data differently according to the 'N' setting on the flow-capture statement. However, there is a bug in the flow-tools documentation such that the default value is truly '3' and not '0' as indicated. The default has been set to $N = 3 to reflect the more common setting. The directory structure associated with $N = 3 is shown below: /flows/router_1/2006/2006-07/2006-07-04 Setting $N=0, would cause the data to accumulate into a single directory without any subdirectories for date organization.	$N = 3;
$use_NDBM	FlowViewer offers the ability to resolve *netflow* IP addresses into their host names on the fly. This process is speeded up by caching names into a 'names' file which resides in the directory specified by the 'names_directory' parameter. As you are building up your 'names' file with early runs, you will notice the speed increase dramatically as the 'names' file is used more. The process of resolving names is the primary reason for slower overall FlowViewer performance. You should preferably use the GDBM array database which is fastest. However, not all Perl distributions support GDBM but most do support NDBM. The '$use_NDBM' flag will cause the FlowViewer_Main.cgi and FlowGrapher_Main.cgi scripts to use NDBM.	$use_NDBM = "N";
$pie_chart_default	The parameter defines which pie-chart option appears as the default on the Pie Chart pulldown on the FlowViewer input screen. The “With Others” option means that the Pie Chart will show an “Others” slice which includes “everything else”. The “Without Others” option will not show an “everything else” slice.	$pie_chart_default = 0; # 0 = None; 1 = With Others; 2 = Without Others
$number_slices	Defines the number of slices included in the Pie Chart.	$number_slices = 6;
$maximum_days	This parameter defines a maximum number of days for the length of user created FlowViewer reports and FlowGrapher graphs.	$maximum_days = 91;
$remove_workfiles_time	This parameter defines the age at which to remove intermediate files from the $flow_working directory when running the FlowViewer_CleanFile utility from *crontab*. (In seconds- the example shows 1 day.)	$remove_workfiles_time = 86400;
$remove_graphfiles_time	This parameter defines the age at which to remove intermediate files from the $graphs_directory directory when running the FlowViewer_CleanFile utility from *crontab*. (In seconds- the example shows 7 days.)	$remove_graphfiles_time = 7*86400;
$remove_reportfiles_time	This parameter defines the age at which to remove intermediate files from the $reports	$remove_reportfiles_time = 7*86400;

	directory when running the FlowViewer_CleanFile utility from *crontab*. (In seconds- the example shows 7 days.)	
$time_zone	This parameter controls the display of the time zone labels for reports, graphs and trackings. Leaving this blank, will result in all labels showing the system time zone (i.e., whatever comes back from ‘timelocal’.)	$time_zone = "EST";
$labels_in_titles	This parameter controls whether to display the Tracking title in the title of the graph itself. Setting this to “1” will include titles, setting it to “0” will not.	$labels_in_titles = "1";
$debug_viewer	This parameter, if set to “Y”, will turn on debugging for FlowViewer. The debug output can be found in $flow_working/DEBUG_VIEWER.	$debug_viewer = “Y”;
$debug_grapher	This parameter, if set to “Y”, will turn on debugging for FlowGrapher. The debug output can be found in $flow_working/DEBUG_GRAPHER.	$debug_grapher = “Y”;
$debug_tracker	This parameter, if set to “Y”, will turn on debugging for FlowTracker. The debug output can be found in $flow_working/DEBUG_TRACKER.	$debug_tracker = “Y”;
$debug_group	This parameter, if set to “Y”, will turn on debugging for FlowTracker_Group. The debug output can be found in $flow_working/DEBUG_GROUP.	$debug_group = “Y”;
$debug_files	This parameter controls whether to save intermediate files for debugging purposes. A value of “Y” will leave the files around for inspection. This defaults to “N”.	$debug_files = “N”;
$log_directory	The location for the logging output files. Some of the logging files, when set to full logging, can get big. Also, if you want the files around for a while, don’t place them in a directory that will get cleaned by one of the FlowViewer_Clean scripts.	$log_directory = "/htp/cgi-bin/FlowViewer_3.3";
$log_collector_short	Provides for a minimal amount of logging for FlowTracker_Collector. A timer is printed which tells how long it has taken to collect the data. This might be useful if you have a lot of Trackings and you want to see if they are still being completed in a timely manner.	$log_collector_short= "Y";
$log_collector_med	Provides for a medium amount of logging for FlowTracker_Collector. A timer is printed which tells how long it has taken to collect the data. This might be useful if you have a lot of Trackings and you want to see if they are still being completed in a timely manner.	$log_collector_med= "N";
$log_collector_long	Provides for a full amount of logging for FlowTracker_Collector. This includes collected data for each active tracking. A timer is printed which tells how long it has taken to collect the data. This might be useful if you have a lot of Trackings and you want to see if they are still being completed in a timely manner.	$log_collector_long= "N";

$log_grapher_short	Provides for a medium amount of logging for FlowTracker_Grapher. The logs have timers showing how long it takes to complete the graphs (e.g., usually under 1 second per tracking).	$log_grapher_short= "Y";
$log_grapher_long	Provides for a full amount of logging for FlowTracker_Grapher. This includes graph data for each active tracking. The logs have timers showing how long it takes to complete the graphs (e.g., usually under 1 second per tracking).	$log_grapher_long= "N";
$collection_offset	Defines how many minutes into the past you want to use to collect data. At 1800 (30 minutes) this will cause FlowTracker_Collector to examine a period 30 minutes in the past. This is useful for allowing all flows that may have crossed that period to be exported from the device. Some flows can last 30 minutes and will be excluded for consideration if they haven’t been exported yet.	$collection_offset = 1800;
$collection_period	This parameter controls how often data is collected for Trackings by FlowTracker_Collector. I have not really used anything other than 5 minutes, so other values have not been tested and YMMV.	$collection_period = 300;
$graphing_period	Frequency at which FlowTracker_Grapher is executed to generate new Tracking graphs.	$graphing_period = 300;
$use_existing_concats	When set to “Y” this parameter will cause FlowTracker_Collector to re-use concatenated flow-tools files for different trackings that are based on the same device. This dramatically speeds things up.	$use_existing_concats = "Y";
$rrd_dir_perms	Controls the UNIX permissions applied to directories of the type defined by the parameter.	$rrd_dir_perms = 0777;
$filter_dir_perms	Controls the UNIX permissions applied to directories of the type defined by the parameter.	$filter_dir_perms = 0777;
$work_dir_perms	Controls the UNIX permissions applied to directories of the type defined by the parameter.	$work_dir_perms = 0777;
$html_dir_perms	Controls the UNIX permissions applied to directories of the type defined by the parameter.	$html_dir_perms = 0777;
$html_file_perms	Controls the UNIX permissions applied to files of the type defined by the parameter.	$html_file_perms = 0777;
$graph_file_perms	Controls the UNIX permissions applied to files of the type defined by the parameter.	$graph_file_perms = 0777;
$rrd_file_perms	Controls the UNIX permissions applied to files of the type defined by the parameter.	$rrd_file_perms = 0777;
$filter_file_perms	Controls the UNIX permissions applied to files of the type defined by the parameter.	$filter_file_perms = 0777;
$tracker_file_perms	Controls the UNIX permissions applied to files of the type defined by the parameter.	$tracker_file_perms = 0777;

$actives_file_perms	Controls the UNIX permissions applied to files of the type defined by the parameter.	$actives_file_perms = 0777;
$saved_filters_perms	Controls the UNIX permissions applied to files of the type defined by the parameter.	$saved_filters_perms= 0777;
$bg_color	Background color of the displayed web pages.	$bg_color = "#F8F8F8";
$text_color	Color of all text appearing on web pages.	$text_color = "#000000";
$link_color	Color of unvisited hyperlinks	$link_color = "#006699";
$vlink_color	Color of visited hyperlinks.	$vlink_color = "#BF294D";
$dig	This parameter points to the location of DNS utility 'dig' (set this to nslookup if required.) The parameter should be set to do inverse DNS lookups, hence the –x in the example.	$dig = "/usr/bin/dig +time=1 -x ";

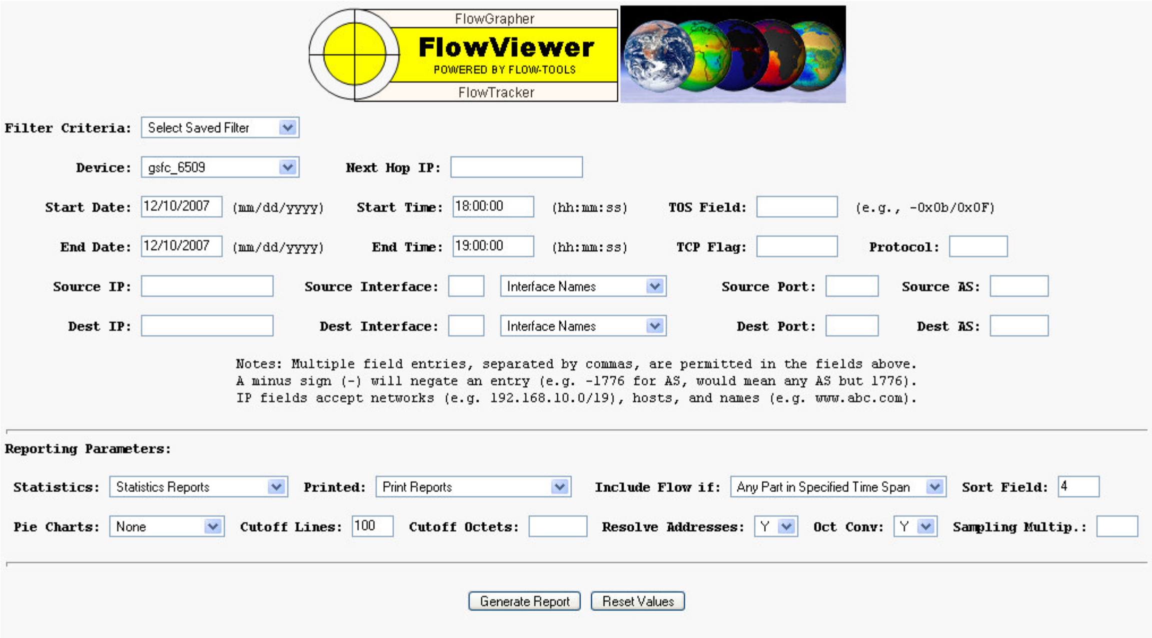
6. FlowViewer Operation and Usage

FlowViewer consists of two parts: FlowViewer.cgi, and FlowViewer_Main.cgi. The user invokes FlowViewer by pointing his browser at the FlowViewer.cgi script. This approach is different from earlier versions and provides the added benefit of updating input date and time periods automatically. Once the user has clicked on the Generate Report button, the FlowViewer_Main.cgi script is invoked which runs several flow-tools to generate the report. The execution of the flow-tools is as follows:

flow-cat
flow-nfilter
flow-print (or)
flow-stat

The parameters for each of these commands are derived from the user's input, including filtering criteria and report selection. The filtering criteria are collected and a used to create a flow-filter file which is provided to flow-nfilter. The script captures the output from either flow-stat or flow-print and formats it for web-page output.

The FlowViewer input screen (FlowViewer.cgi) is shown in figure 6-1 below:



Filter Criteria: Select Saved Filter

Device: gsfc_6509 Next Hop IP:

Start Date: 12/10/2007 (mm/dd/yyyy) Start Time: 18:00:00 (hh:mm:ss) TOS Field: (e.g., -0x0b/0x0F)

End Date: 12/10/2007 (mm/dd/yyyy) End Time: 19:00:00 (hh:mm:ss) TCP Flag: Protocol:

Source IP: Source Interface: Interface Names Source Port: Source AS:

Dest IP: Dest Interface: Interface Names Dest Port: Dest AS:

Notes: Multiple field entries, separated by commas, are permitted in the fields above.
A minus sign (-) will negate an entry (e.g. -1776 for AS, would mean any AS but 1776).
IP fields accept networks (e.g. 192.168.10.0/19), hosts, and names (e.g. www.abc.com).

Reporting Parameters:

Statistics: Statistics Reports Printed: Print Reports Include Flow if: Any Part in Specified Time Span Sort Field: 4

Pie Charts: None Cutoff Lines: 100 Cutoff Octets: Resolve Addresses: Y Oct Conv: Y Sampling Multip.:

Figure 6-1 FlowViewer input screen

The user will complete input fields as necessary to define a filter for viewing flow_data. Or, as of version 3.3 the user may select from a previously defined filter. Filters are saved for future reference after a report has been produced by clicking on the “Save Filter” link embedded in the FlowViewer_Save.png logo.

FlowViewer will accept up to 10 entries for each field, separated by commas. Fields may be preceded by a dash or minus sign (-), which will cause the script to ignore such flows. For example, providing the value -1776 to the Source AS field will eliminate from the report any flows that originated in AS 1776.

The “Device Name” field allows the user to select from a collection of devices that he may be collecting netflow from (‘@devices field in the FlowViewer_Configuration.pm file.) As of version 3.3, the user is also able to differentiate netflow data from different exporters based on the ‘Exporter ID’ field. This is used in the situation where the user is collecting from multiple devices onto the same *flow-capture* port. The Exporter field is not shown above, but appears if the user has configured the ‘@exporters’ field in the FlowViewer_Configuration.pm file.

If you are not using any devices or exporters, you will have to set:

$no_devices_or_exporters = “Y”;

In this case, no device or exporter pulldown will appear.

The Source and Destination IP fields can accept either individual IP addresses, network base and range (e.g., 192.169.100.0/24), or fully qualified domain names (e.g., www.abccompany.com.)

The Source Port and Destination Port entry fields will accept a range value as of version 3.1. The range value is created by separating the range end values with a colon (e.g., 40100:40200.) Note that the underlying software, flow-tools, does not provide a range capability. FlowViewer mimics the capability by individually listing each value within the range. This could make for a very long filtering line to be provided to flow-tools. The performance for very long ranges is not known.

The Source and Destination Interface values expect the SNMP index value for the device's interfaces. Note that these can change over time (e.g., when a new interface card is added to the device.) For FlowTracker this becomes important. If an interface index value should change for an active Tracking, use the 'Revise' option on the FlowTracker main page to modify the filter. This will maintain the integrity of the Tracking. As of version 3.3, FlowViewer offers the option to use Named Interfaces. These must be configured in advance in either the NamedInterfaces_Devices, or NamedInterfaces_Exporters files. Only one interface is available via the NamedInterfaces pulldown, however this may be combined with numeric values in the original interfaces text box to filter on multiple interfaces.

After completing the Filter Criteria, the user selects either a Statistics Report, or a Printed Report. FlowViewer provides a web page of the same output that the flow-tools report generates from the command line. The current reports that are available include:

Statistical Reports:

Summary
UDP/TCP Destination Port
UDP/TCP Source Port
UDP/TCP Port
Destination IP
SourceIP
Source/Destination IP
Source or Destination IP
IP Protocol
Input Interface
Output Interface
Input/Output Interface
Source AS
Destination AS
Source/Destination AS
IP ToS
Source Prefix
Destination Prefix
Source/Destination Prefix

Printed Reports:

Flow Times
AS Numbers
132 Columns
1 Line with Tags
AS Aggregation

Protocol Port Aggregation
Source Prefix Aggregation
Destination Prefix Aggregation
Prefix Aggregation
Full (Catalyst)

The “Include Flow If” parameter allows the user several options for controlling which flows are included in the report. Because flows do not completely lie within a specified period, the user has the option to define the conditions for including the flow. These include:

Any Part in Specified Time Span
End Time in Specified Time Span
Start Time in Specified Time Span
Entirely in Specified Time Span

The “Sort Field” parameter controls the ordering of the report based on which column has been selected for sorting. You can precede the sort field value with a dash (-) to specify a reverse sorting order.

The user now has the option to view FlowViewer results in text form together with a pie-chart representation of the data. The user would select a particular “Pie Charts” option.

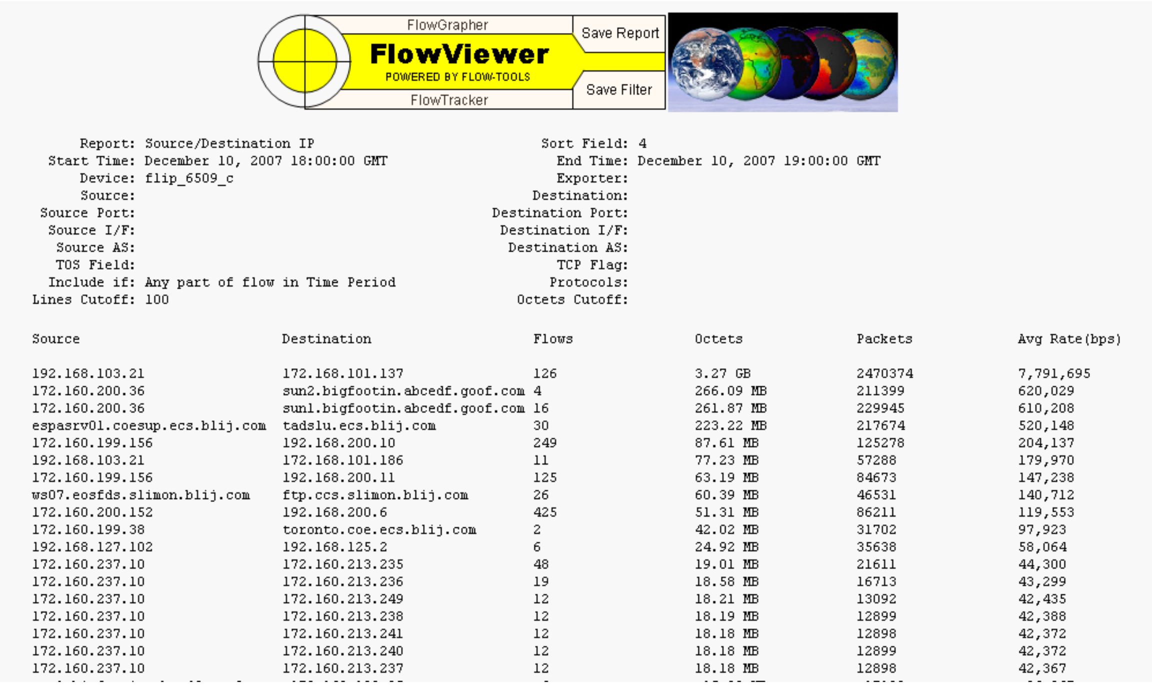
The “Cutoff Lines” parameter controls how many lines will be printed. The first ‘cutoff_lines’ are printed. The ’Cutoff Octets’ parameter controls the point at which to end the report based on the number of octets displayed in the output line. No additional lines will be printed which contain an Octets value less than ‘cutoff octets’.

The “Sampling Multiplier” field can be set to a value greater than one whereby the data in all reports and graphs will be multiplied by this number. This field is used to give an approximation of real traffic flow levels for devices that export sampled netflow data.

The “Oct Conv.” option if selected will display octets in a shorthand notation (e.g., 10.3 MB instead of 10300000.)

The “Resolve Addresses” parameter informs the script whether or not to resolve IP addresses into their full host names. Resolving addresses is a little slow the first time through, but builds up a cache as the number of runs increases and soon becomes as fast as not resolving addresses.

A typical FlowViewer report (in this case Input/Output Interface) is shown in figure 6-2 below:



```
     Report: Source/Destination IP                         Sort Field: 4
 Start Time: December 10, 2007 18:00:00 GMT                  End Time: December 10, 2007 19:00:00 GMT
     Device: flip_6509_c                                     Exporter:
     Source:                                              Destination:
Source Port:                                         Destination Port:
 Source I/F:                                          Destination I/F:
  Source AS:                                           Destination AS:
  TOS Field:                                                 TCP Flag:
 Include if: Any part of flow in Time Period                Protocols:
Lines Cutoff: 100                                      Octets Cutoff:

Source                         Destination                    Flows               Octets              Packets             Avg Rate(bps)

192.168.103.21                 172.168.101.137                126                 3.27 GB             2470374             7,791,695
172.160.200.36                 sun2.bigfootin.abcedf.goof.com 4                   266.09 MB           211399              620,029
172.160.200.36                 sun1.bigfootin.abcedf.goof.com 16                  261.87 MB           229945              610,208
espasrv01.coesup.ecs.blij.com  tadslu.ecs.blij.com            30                  223.22 MB           217674              520,148
172.160.199.156                192.168.200.10                 249                 87.61 MB            125278              204,137
192.168.103.21                 172.168.101.186                11                  77.23 MB            57288               179,970
172.160.199.156                192.168.200.11                 125                 63.19 MB            84673               147,238
ws07.eosfds.slimon.blij.com    ftp.ccs.slimon.blij.com        26                  60.39 MB            46531               140,712
172.160.200.152                192.168.200.6                  425                 51.31 MB            86211               119,553
172.160.199.38                 toronto.coe.ecs.blij.com       2                   42.02 MB            31702               97,923
192.168.127.102                192.168.125.2                  6                   24.92 MB            35638               58,064
172.160.237.10                 172.160.213.235                48                  19.01 MB            21611               44,300
172.160.237.10                 172.160.213.236                19                  18.58 MB            16713               43,299
172.160.237.10                 172.160.213.249                12                  18.21 MB            13092               42,435
172.160.237.10                 172.160.213.238                12                  18.19 MB            12899               42,388
172.160.237.10                 172.160.213.241                12                  18.18 MB            12898               42,372
172.160.237.10                 172.160.213.240                12                  18.18 MB            12899               42,372
172.160.237.10                 172.160.213.237                12                  18.18 MB            12898               42,367
```

Figure 6-2 FlowViewer report output

From the FlowViewer report output page the user can ‘Save Report’, or ‘Save Filter’ via links in the FlowViewer logo at the top of the page. Saved reports listed on an Saved Reports page as of version 3.3. Filters can now be saved for future retrieval by either of FlowViewer, FlowGrapher, or FlowTracker.

FlowViewer Tips

- The 132 Column Printed Report option is very useful for understanding flows through your network. The report provides source and destination information and interface and port information in the same output. This length of this report is constrained primarily by the ‘Cutoff Lines’ parameter, but is not slowed down by large values.

- Some reports will not work unless the proper netflow export field has been collected.

- The Input and Output interfaces are represented by the SNMP index assigned to each interface by the device. In version 3.3 these will be named if NamedInterfaces have been configured for the device or exporter selected.

7. FlowGrapher Operation and Usage

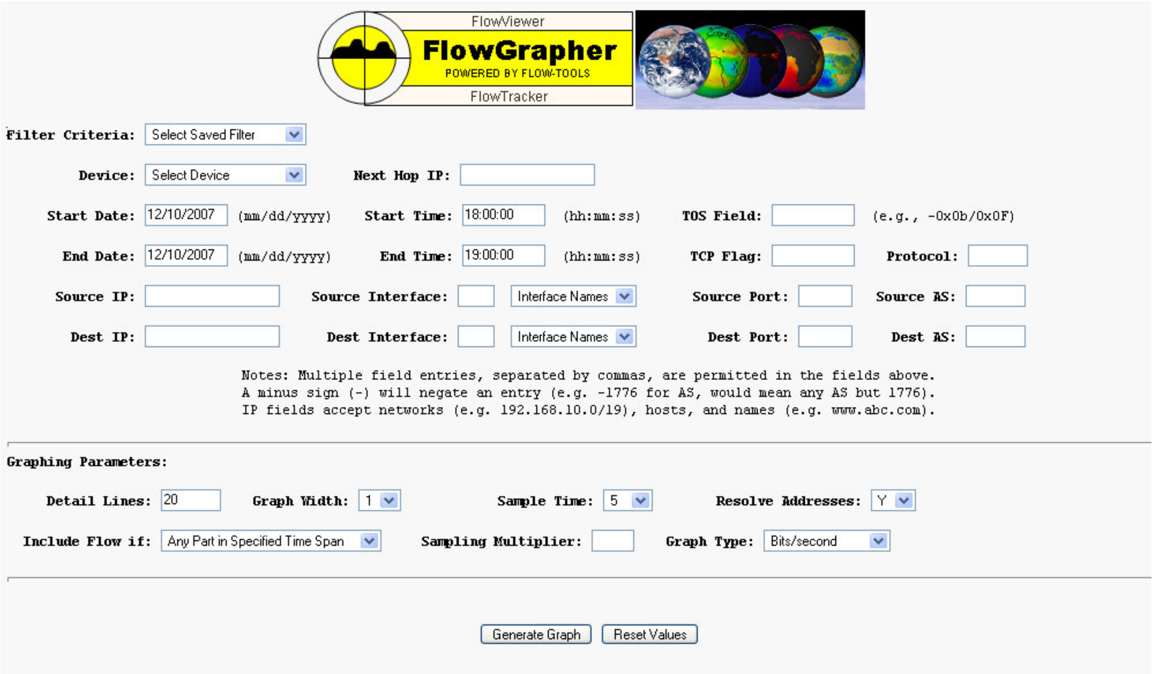
FlowGrapher consists of two parts: FlowGrapher.cgi, and FlowGrapher_Main.cgi. The user invokes FlowGrapher by pointing his browser at the FlowGrapher.cgi script. This approach is different from earlier versions and provides the added benefit of updating input date and time periods automatically. Once the user has clicked on the Generate Graph button, the FlowGrapher_Main.cgi script is invoked which runs several flow-tools to generate the report. The execution of the flow-tools is as follows:

flow-cat
flow-nfilter
flow-print (132 columns)

The parameters for each of these commands are derived from the user's input, primarily the filtering criteria. The filtering criteria are collected and a used to create a flow-filter file which is provided to flow-nfilter. The script captures the output from the flow-print 132-columns option and parses it to build the graph.

The script builds an array of times and values depending on the "Sample Time" parameter (in seconds.) This parameter defines the width of the 'buckets' into which segments of flow-data is accumulated. When all of the 132-column output has been parsed, the array is provided to Lincoln Stein's GD::Graph software which produces the graph.

The FlowGrapher input screen (FlowGrapher.cgi) is shown in figure 7-1 below:



Filter Criteria: Select Saved Filter

Device: Select Device   Next Hop IP:

Start Date: 12/10/2007 (mm/dd/yyyy)   Start Time: 18:00:00 (hh:mm:ss)   TOS Field: (e.g., -0x0b/0x0F)

End Date: 12/10/2007 (mm/dd/yyyy)   End Time: 19:00:00 (hh:mm:ss)   TCP Flag: Protocol:

Source IP: Source Interface: Interface Names   Source Port: Source AS:

Dest IP: Dest Interface: Interface Names   Dest Port: Dest AS:

Notes: Multiple field entries, separated by commas, are permitted in the fields above.
A minus sign (-) will negate an entry (e.g. -1776 for AS, would mean any AS but 1776).
IP fields accept networks (e.g. 192.168.10.0/19), hosts, and names (e.g. www.abc.com).

Graphing Parameters:

Detail Lines: 20   Graph Width: 1   Sample Time: 5   Resolve Addresses: Y

Include Flow if: Any Part in Specified Time Span   Sampling Multiplier: Graph Type: Bits/second

Figure 7-1 FlowGrapher input screen

The user will complete input fields as necessary to define a filter for limiting the flow_data. The filtering criteria are identical to those from FlowViewer described above.

The “Detail Lines” parameter controls how many lines of flow detail information will be printed. FlowGrapher will select the largest ‘detail_lines’ number of flows to present below the graph.

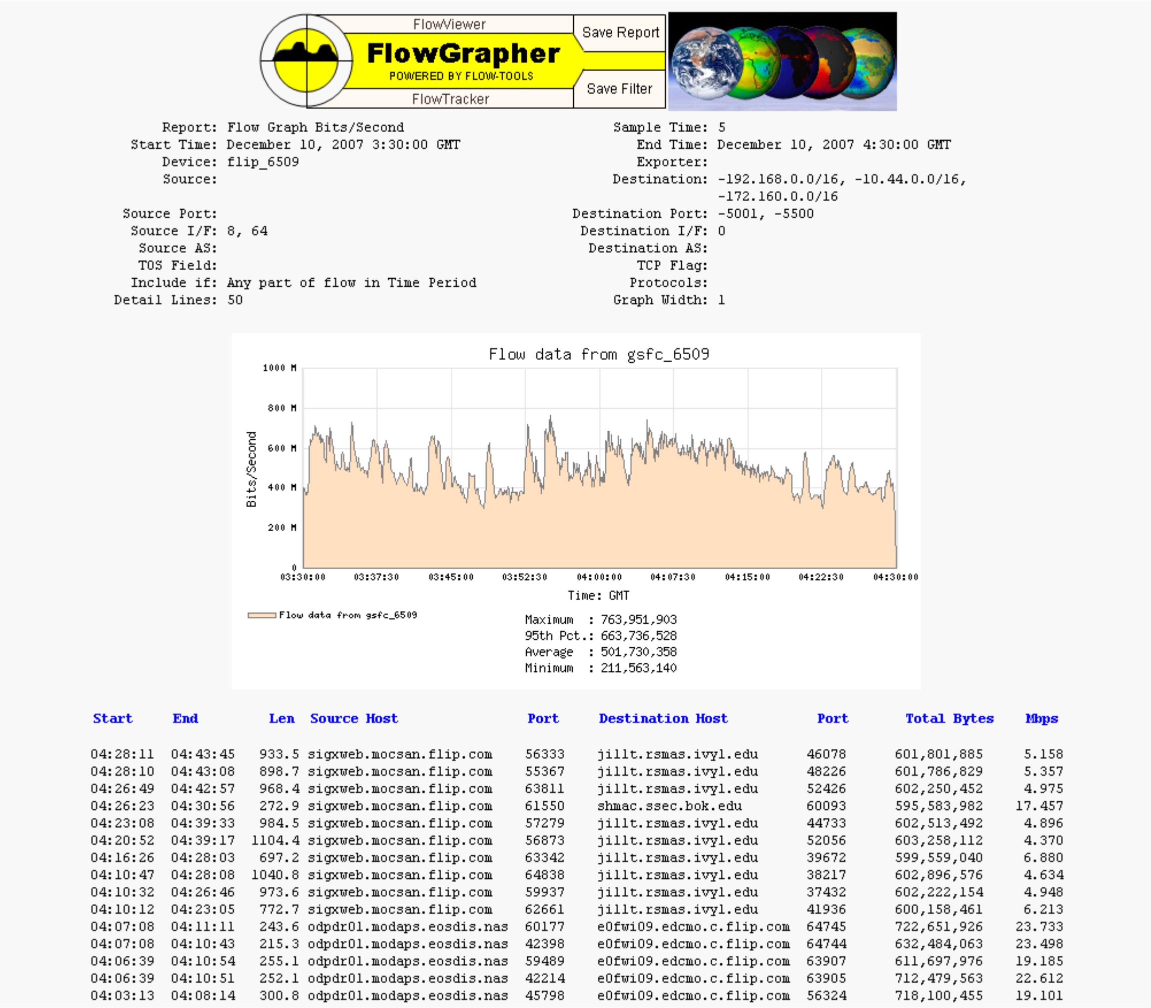
The “Graph Width” parameter is used to scale the resulting graph image. This is useful sometimes for viewing detailed graphs.

The “Resolve Addresses” and “Include Flow If” parameters are the same as with FlowViewer, and are described above.

The “Graph Types” option allows the user to graph either octets, flows, or packets.

The “Sampling Multiplier” field allows the user to expand the graphed output in compliance with the sampling rate for sampled netflow data in order to simulate actual traffic flows.

A typical FlowGrapher report is shown in figure 7-2 below:



```
       Report: Flow Graph Bits/Second                      Sample Time: 5
   Start Time: December 10, 2007 3:30:00 GMT                  End Time: December 10, 2007 4:30:00 GMT
       Device: flip_6509                                      Exporter:
       Source:                                             Destination: -192.168.0.0/16, -10.44.0.0/16,
                                                                        -172.160.0.0/16
  Source Port:                                        Destination Port: -5001, -5500
   Source I/F: 8, 64                                   Destination I/F: 0
    Source AS:                                          Destination AS:
    TOS Field:                                                TCP Flag:
   Include if: Any part of flow in Time Period               Protocols:
 Detail Lines: 50                                          Graph Width: 1
```

```
Maximum  : 763,951,903
95th Pct.: 663,736,528
Average  : 501,730,358
Minimum  : 211,563,140
```

```
Start     End          Len  Source Host                  Port     Destination Host          Port       Total Bytes    Mbps

04:28:11  04:43:45   933.5 sigxweb.mocsan.flip.com      56333    jillt.rsmas.ivyl.edu      46078     601,801,885    5.158
04:28:10  04:43:08   898.7 sigxweb.mocsan.flip.com      55367    jillt.rsmas.ivyl.edu      48226     601,786,829    5.357
04:26:49  04:42:57   968.4 sigxweb.mocsan.flip.com      63811    jillt.rsmas.ivyl.edu      52426     602,250,452    4.975
04:26:23  04:30:56   272.9 sigxweb.mocsan.flip.com      61550    shmac.ssec.bok.edu        60093     595,583,982   17.457
04:23:08  04:39:33   984.5 sigxweb.mocsan.flip.com      57279    jillt.rsmas.ivyl.edu      44733     602,513,492    4.896
04:20:52  04:39:17  1104.4 sigxweb.mocsan.flip.com      56873    jillt.rsmas.ivyl.edu      52056     603,258,112    4.370
04:16:26  04:28:03   697.2 sigxweb.mocsan.flip.com      63342    jillt.rsmas.ivyl.edu      39672     599,559,040    6.880
04:10:47  04:28:08  1040.8 sigxweb.mocsan.flip.com      64838    jillt.rsmas.ivyl.edu      38217     602,896,576    4.634
04:10:32  04:26:46   973.6 sigxweb.mocsan.flip.com      59937    jillt.rsmas.ivyl.edu      37432     602,222,154    4.948
04:10:12  04:23:05   772.7 sigxweb.mocsan.flip.com      62661    jillt.rsmas.ivyl.edu      41936     600,158,461    6.213
04:07:08  04:11:11   243.6 odpdr01.modaps.eosdis.nas    60177    e0fwi09.edcmo.c.flip.com  64745     722,651,926   23.733
04:07:08  04:10:43   215.3 odpdr01.modaps.eosdis.nas    42398    e0fwi09.edcmo.c.flip.com  64744     632,484,063   23.498
04:06:39  04:10:54   255.1 odpdr01.modaps.eosdis.nas    59489    e0fwi09.edcmo.c.flip.com  63907     611,697,976   19.185
04:06:39  04:10:51   252.1 odpdr01.modaps.eosdis.nas    42214    e0fwi09.edcmo.c.flip.com  63905     712,479,563   22.612
04:03:13  04:08:14   300.8 odpdr01.modaps.eosdis.nas    45798    e0fwi09.edcmo.c.flip.com  56324     718,100,455   19.101
```

Figure 7-2 FlowGrapher output

As of version 3.1, FlowGrapher now display statistical information about the data flows for the time period graphed. The information includes the maximum, minimum, average, and 95th percentile values of those data points plotted. As of version 3.3, each of the data lines are sortable. To sort, click on the column header that you wish to sort by.

FlowGrapher Tips

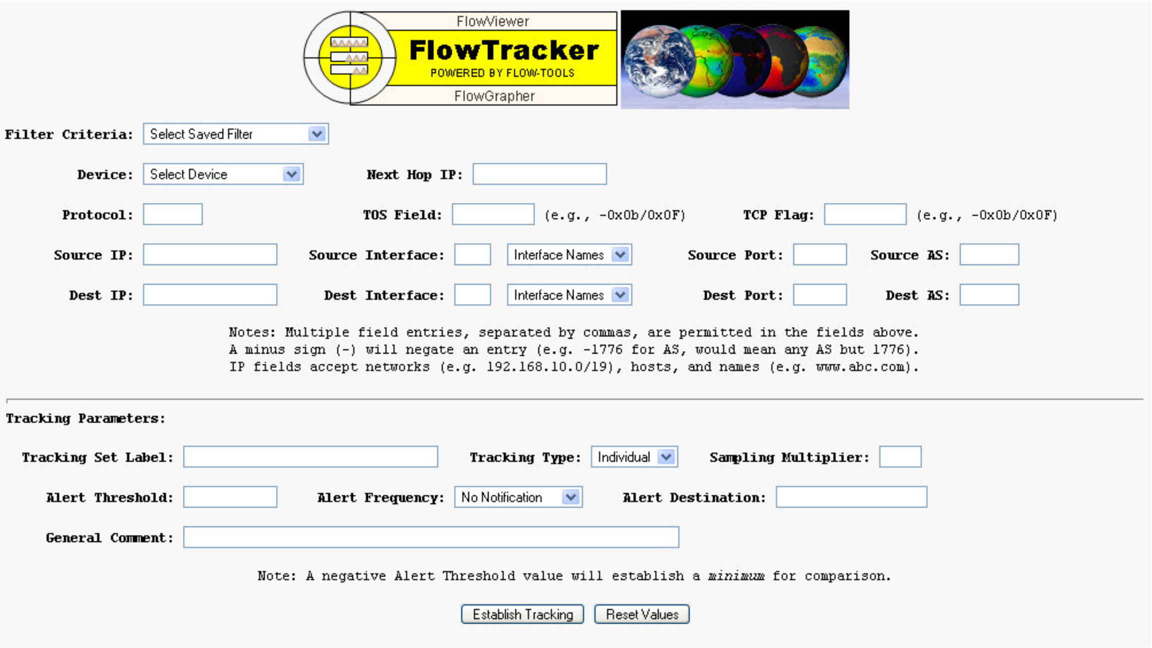
- FlowGrapher completion speed is effected by the ‘Detail Lines’ input variable. FlowGrapher will select the largest ‘Detail Lines’ number of flows to display. For example the largest 100 flows. Sorting the largest flows can slow down for very large values of ‘Detail Lines.’

- Varying the ‘Sample Time’ parameter (which effects the size of the bucket into which flows are parsed) does not have a significant impact on report completion speed.

8. FlowTracker Operation and Usage

FlowTracker consists of three parts: FlowTracker.cgi, FlowTracker_Main.cgi, and FlowTracker_Group.cgi. The user invokes FlowTracker by pointing his browser at the FlowTracker.cgi script. The FlowTracker input screen offers the same basic filtering criteria, with the exception that there in no longer the capability to enter start and end times. This is because Trackings established by FlowTracker are updated continuously and the time period is established by the FlowTracker_Collector program. This script runs every five minutes and extracts flow data amounts, based on the established Tracking filter, for a 5-minute period approximately 30 minutes in the past.

The FlowTracker input screen (figure 8-1 below) permits the user to define a filter for a long term Tracking or to create a Group from pre-existing trackings. The user is prompted to provide a Tracking name and to supply any comments that might help explain the Tracking.



Filter Criteria: Select Saved Filter

Device: Select Device   Next Hop IP:

Protocol: TOS Field: (e.g., -0x0b/0x0F)   TCP Flag: (e.g., -0x0b/0x0F)

Source IP: Source Interface: Interface Names   Source Port: Source AS:

Dest IP: Dest Interface: Interface Names   Dest Port: Dest AS:

Notes: Multiple field entries, separated by commas, are permitted in the fields above.
A minus sign (-) will negate an entry (e.g. -1776 for AS, would mean any AS but 1776).
IP fields accept networks (e.g. 192.168.10.0/19), hosts, and names (e.g. www.abc.com).

Tracking Parameters:

Tracking Set Label: Tracking Type: Individual   Sampling Multiplier:

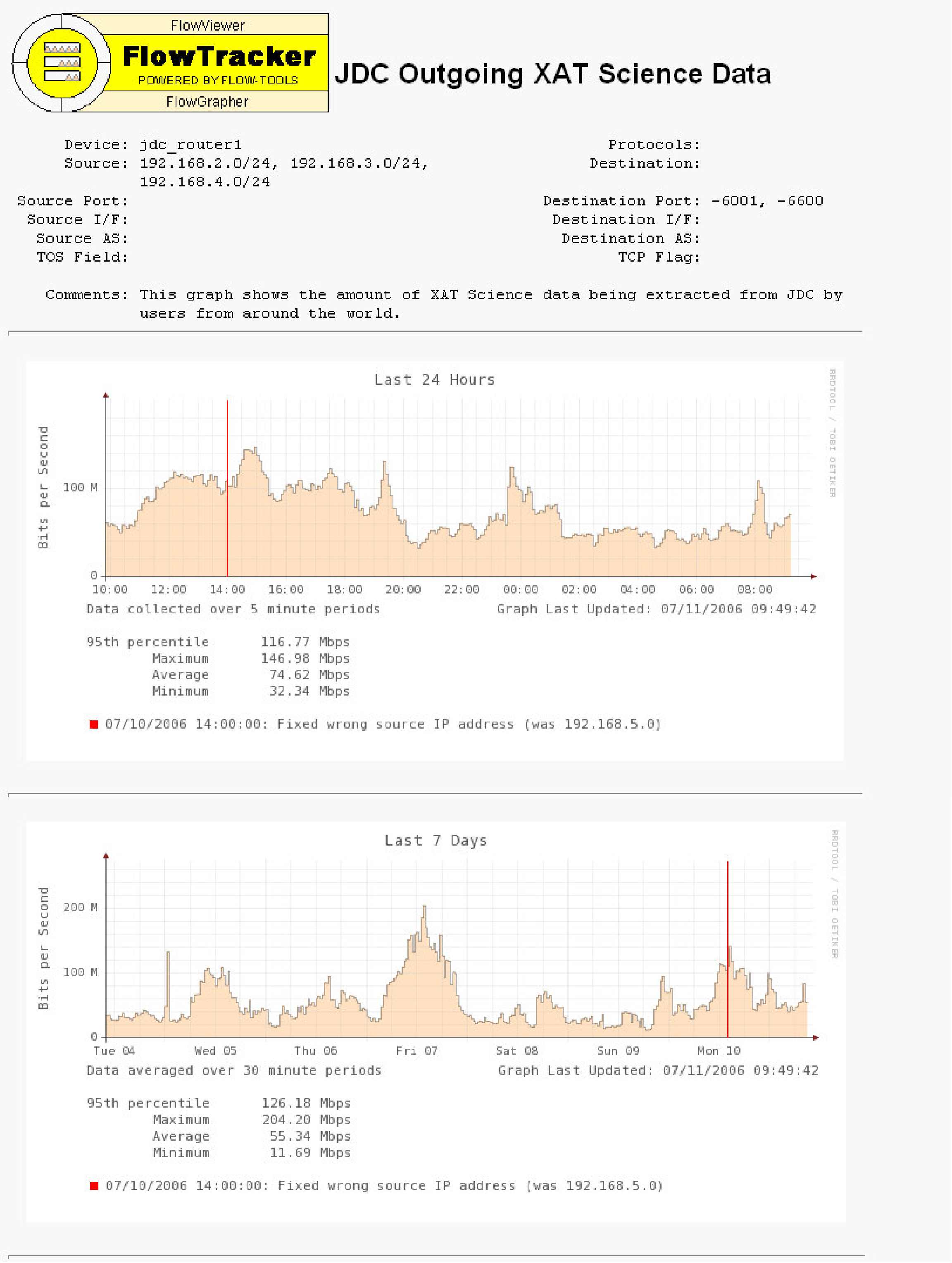
Alert Threshold: Alert Frequency: No Notification   Alert Destination:

General Comment:

Note: A negative Alert Threshold value will establish a *minimum* for comparison.

Figure 8-1 FlowTracker Input Screen

A Tracking is filter-driven and basically results in four MRTG-like graphs which track flow amounts over four time periods; daily, weekly, monthly, and yearly. A sample Tracking is shown in figure 8-2 below. A tracking group is established from existing trackings and is invoked when the user selects ‘Group’ for the Tracking Type. A group does not have any input filtering criteria and any supplied will be ignored. A group will also have the four MRTG-like graphs created from the stacking of values from each of the group’s components. The Group Tracking input screen is discussed in more detail below.



JDC Outgoing XAT Science Data

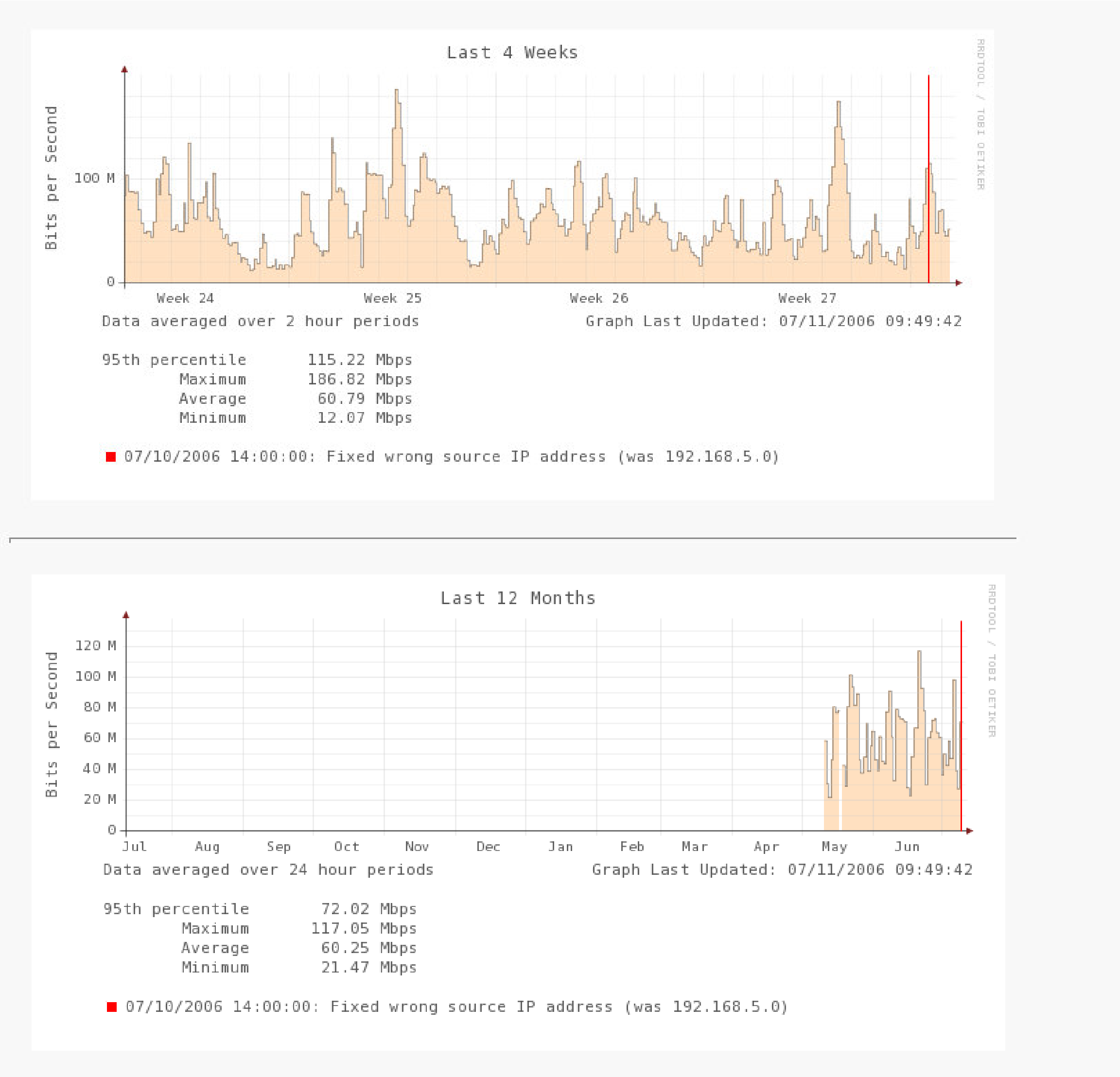
```
     Device: jdc_router1                                Protocols:
     Source: 192.168.2.0/24, 192.168.3.0/24,          Destination:
             192.168.4.0/24
Source Port:                                   Destination Port: -6001, -6600
 Source I/F:                                    Destination I/F:
  Source AS:                                     Destination AS:
  TOS Field:                                           TCP Flag:

   Comments: This graph shows the amount of XAT Science data being extracted from JDC by
             users from around the world.
```

Last 24 Hours

```
Data collected over 5 minute periods          Graph Last Updated: 07/11/2006 09:49:42

95th percentile      116.77 Mbps
        Maximum      146.98 Mbps
        Average       74.62 Mbps
        Minimum       32.34 Mbps

■ 07/10/2006 14:00:00: Fixed wrong source IP address (was 192.168.5.0)
```

Last 7 Days

```
Data averaged over 30 minute periods          Graph Last Updated: 07/11/2006 09:49:42

95th percentile      126.18 Mbps
        Maximum      204.20 Mbps
        Average       55.34 Mbps
        Minimum       11.69 Mbps

■ 07/10/2006 14:00:00: Fixed wrong source IP address (was 192.168.5.0)
```



```
Data averaged over 2 hour periods                Graph Last Updated: 07/11/2006 09:49:42

95th percentile      115.22 Mbps
        Maximum      186.82 Mbps
        Average       60.79 Mbps
        Minimum       12.07 Mbps

■ 07/10/2006 14:00:00: Fixed wrong source IP address (was 192.168.5.0)
```

```
Data averaged over 24 hour periods               Graph Last Updated: 07/11/2006 09:49:42

95th percentile       72.02 Mbps
        Maximum      117.05 Mbps
        Average       60.25 Mbps
        Minimum       21.47 Mbps

■ 07/10/2006 14:00:00: Fixed wrong source IP address (was 192.168.5.0)
```

Figure 8-2 FlowTracker Tracking

It is useful to experiment with different filters in FlowViewer or FlowGrapher before settling on the right filter for creating a Tracking. When you are satisfied, simply click on the FlowTracker link embedded in the FlowViewer or FlowGrapher logo and the filter criteria you finally ended up with will pre-fill the FlowTracker input screen. Although you can make any number of modifications to a tracking once it is created, a maximum of three of these modifications can be applied to the graph itself, creating a vertical line to mark the change. FlowTracker will use the last three notations created.

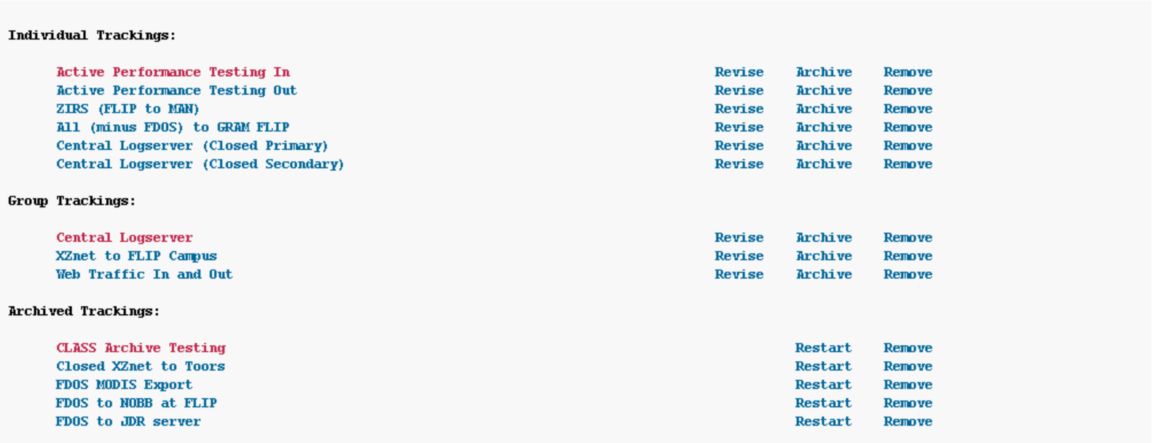
The FlowTracker screen allows a user to create new Trackings, and to manage existing ones. For existing Trackings, the user has the ability to “Revise” or “Remove” them. The “Revise” feature permits the user to adjust an existing Tracking to use a modified filter, or to change the comment associated with a Tracking. The Tracking will continue with these modifications, thus preserving historical data. The Tracking sample below has such a modification.

The Source and Destination Interface values expect the SNMP index value for the device's interfaces. Note that these can change over time (e.g., when a new interface card is added to the device.) For FlowTracker this becomes important. If an interface index value should change for an active Tracking, use the 'Revise' option on the FlowTracker main page to modify the filter. This will maintain the integrity of the Tracking.

The user, as of version 3.3, may establish an Alert Threshold and be alerted via email whenever this threshold has been exceeded; or for negative values, whenever the tracking does not meet the threshold. Fields have been included for this purpose. The user may elect to be notified 'with every occurrence', 'once a day', or to stop notifications.

When the user clicks on the "Establish Tracking" button, the FlowTracker_Main.cgi script is invoked. When the Tracking Type is set to 'Individual', this script creates a filter file to preserve the filter criteria, and an RRDtool database to maintain the 5-minute flow data readings for each Tracking, based on the filter data. The script will also create an HTML page to hold the filter criteria and the four MRTG-like graphs. It will also establish a directory to hold all of the files (i.e., index.html, FlowTracker_Links.png, and the four graphs.)

Existing Trackings and Groups have been listed on the FlowTracker input page for management. Figure 8-3 shows a portion of the page.



```
Individual Trackings:

      Active Performance Testing In                                        Revise    Archive    Remove
      Active Performance Testing Out                                       Revise    Archive    Remove
      ZIRS (FLIP to MAN)                                                   Revise    Archive    Remove
      All (minus FDOS) to GRAM FLIP                                        Revise    Archive    Remove
      Central Logserver (Closed Primary)                                   Revise    Archive    Remove
      Central Logserver (Closed Secondary)                                 Revise    Archive    Remove

Group Trackings:

      Central Logserver                                                    Revise    Archive    Remove
      XZnet to FLIP Campus                                                 Revise    Archive    Remove
      Web Traffic In and Out                                               Revise    Archive    Remove

Archived Trackings:

      CLASS Archive Testing                                                          Restart    Remove
      Closed XZnet to Toors                                                          Restart    Remove
      FDOS MODIS Export                                                              Restart    Remove
      FDOS to NOBB at FLIP                                                           Restart    Remove
      FDOS to JDR server                                                             Restart    Remove
```

Figure 8-3 FlowTracker listing of existing Trackings for management

A user may remove a Tracking. In this case, the script moves the tracking files that were created (see below) to the working directory for deletion later. This allows the user a chance to recover if he has done this by mistake.

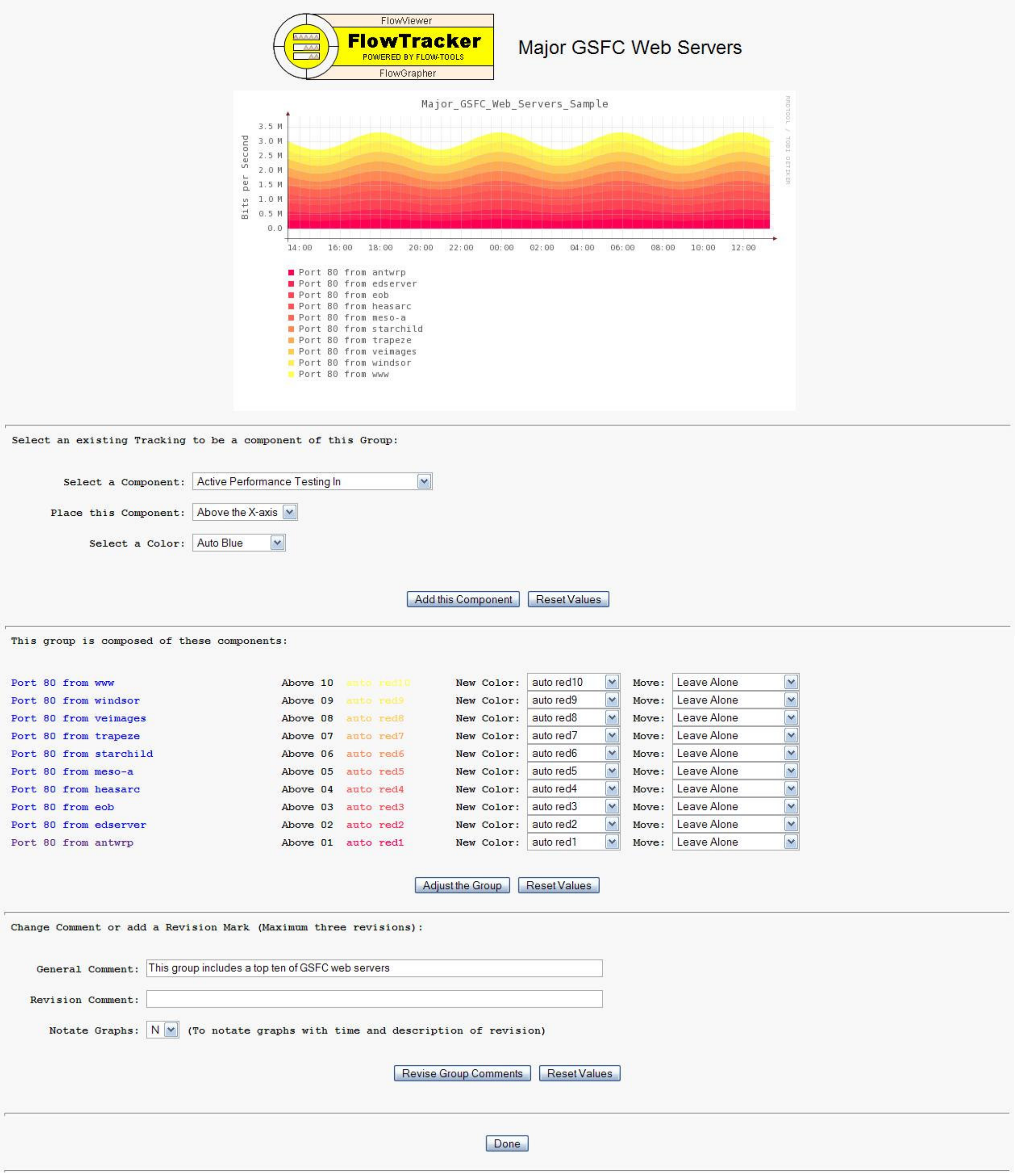
When the user wishes to create a Group from previously defined existing (Individual) trackings, he selects 'Group' from the Tracking Type pulldown. No tracking filter criteria are required and in fact are ignored if provided since a group has no filter criteria or RRDtool databases associated with it directly. The FlowTracker_Main.cgi script will invoke the FlowTracker_Group.cgi script which will handle the user's creation of a group. The Group tracking input screen is shown in figure 8-4 below.

When defining a group, the user identifies which existing tracking he would like to add to the group next. He identifies whether it should be placed above or below the x-axis, and which color it should be. There are four automatic colors (red, green, blue, violet) which if selected will inform the script to automatically use the next color in a range of similar colors. Each time a new component (i.e., a selected existing tracking) is added to the graph, a sample RRDtool graph is created. Note that the RRDs.pm component of the RRDtool distribution is used to speed things up. If you are having problems creating these sample graphs, make sure your RRDs.pm is installed and compatible with your RRDtool version (it usually is.)

The FlowTracker Group input screen allows the user to move components around and change their colors until a satisfactory group is achieved. Groups can be revised just like individual trackings can and the group graph can be notated with a vertical bar for a particular revision if desired.

Once a group is established it will appear with the next execution of FlowTracker_Grapher (defaults to every 5 minutes.) The group will appear in the Group area of the Active Trackings webpage and also in a Group area of the FlowTracker Input screen.

Since version 3.1, users have the ability to archive trackings, and restore them to active collection and graphing later if they wish. This can be useful for removing inactive trackings from both the collecting and graphing processes.



Major GSFC Web Servers

Select an existing Tracking to be a component of this Group:

Select a Component: Active Performance Testing In

Place this Component: Above the X-axis

Select a Color: Auto Blue

This group is composed of these components:

Port 80 from www	Above 10	auto red10	New Color:	auto red10	Move:	Leave Alone
Port 80 from windsor	Above 09	auto red9	New Color:	auto red9	Move:	Leave Alone
Port 80 from veimages	Above 08	auto red8	New Color:	auto red8	Move:	Leave Alone
Port 80 from trapeze	Above 07	auto red7	New Color:	auto red7	Move:	Leave Alone
Port 80 from starchild	Above 06	auto red6	New Color:	auto red6	Move:	Leave Alone
Port 80 from meso-a	Above 05	auto red5	New Color:	auto red5	Move:	Leave Alone
Port 80 from heasarc	Above 04	auto red4	New Color:	auto red4	Move:	Leave Alone
Port 80 from eob	Above 03	auto red3	New Color:	auto red3	Move:	Leave Alone
Port 80 from edserver	Above 02	auto red2	New Color:	auto red2	Move:	Leave Alone
Port 80 from antwrp	Above 01	auto red1	New Color:	auto red1	Move:	Leave Alone

Change Comment or add a Revision Mark (Maximum three revisions):

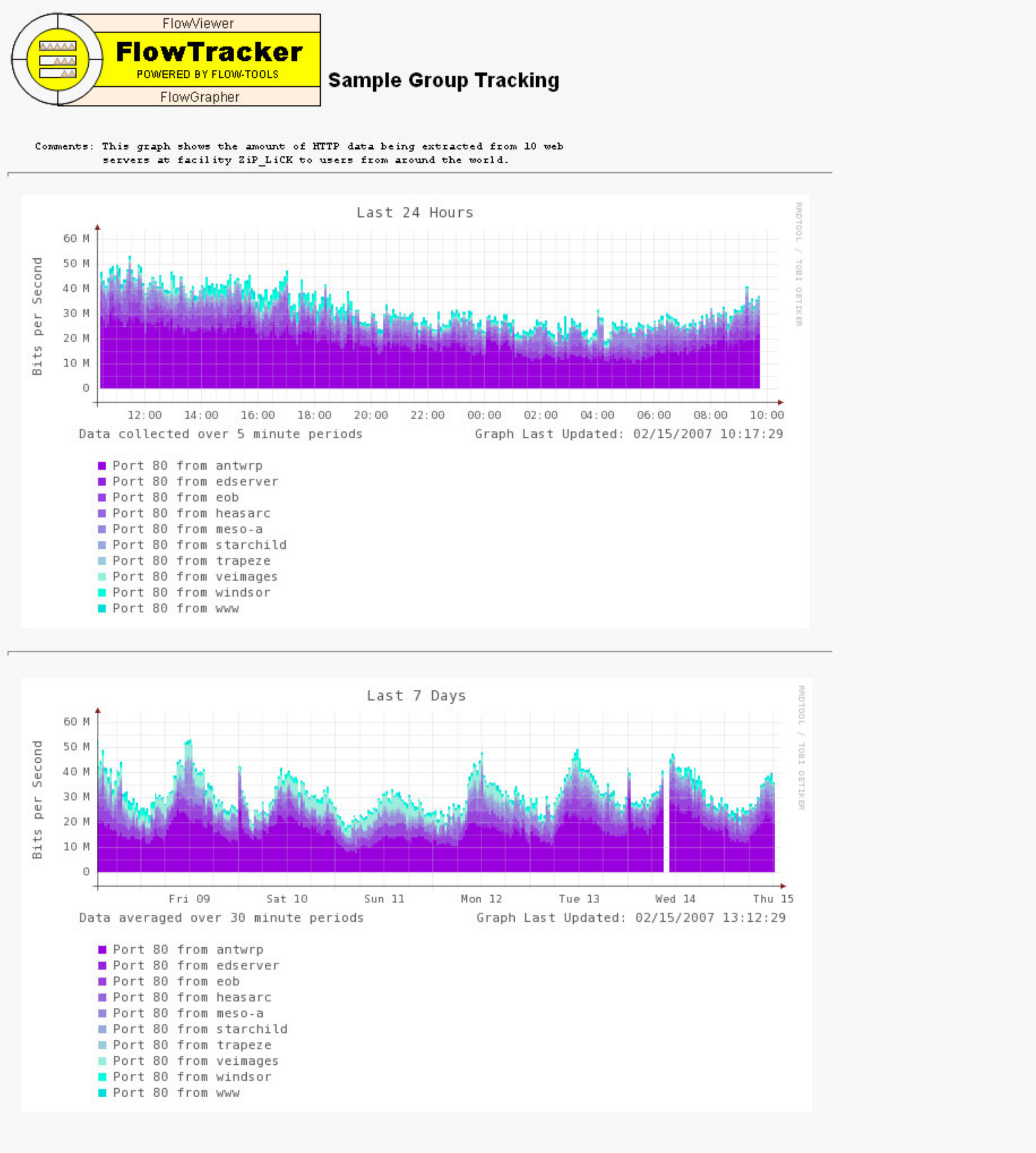
General Comment: This group includes a top ten of GSFC web servers

Revision Comment:

Notate Graphs: N (To notate graphs with time and description of revision)

Figure 8-4 FlowTracker Group Input screen

A typical FlowTracker Group web page is shown in figure 8.5 below. Group trackings have the same ability for revision as individual trackings, with vertical bars placed on the group graphs when the user requests that revisions to the group be notated. Each of the individual trackings that make up the group are listed beneath the graph and each is an embedded link back to the individual tracking web page.



Sample Group Tracking

```
Comments: This graph shows the amount of HTTP data being extracted from 10 web
          servers at facility ZiP_LiCK to users from around the world.
```

Last 24 Hours

Data collected over 5 minute periods   Graph Last Updated: 02/15/2007 10:17:29

- ■ Port 80 from antwrp
- ■ Port 80 from edserver
- ■ Port 80 from eob
- ■ Port 80 from heasarc
- ■ Port 80 from meso-a
- ■ Port 80 from starchild
- ■ Port 80 from trapeze
- ■ Port 80 from veimages
- ■ Port 80 from windsor
- ■ Port 80 from www

Last 7 Days

Data averaged over 30 minute periods   Graph Last Updated: 02/15/2007 13:12:29

- ■ Port 80 from antwrp
- ■ Port 80 from edserver
- ■ Port 80 from eob
- ■ Port 80 from heasarc
- ■ Port 80 from meso-a
- ■ Port 80 from starchild
- ■ Port 80 from trapeze
- ■ Port 80 from veimages
- ■ Port 80 from windsor
- ■ Port 80 from www

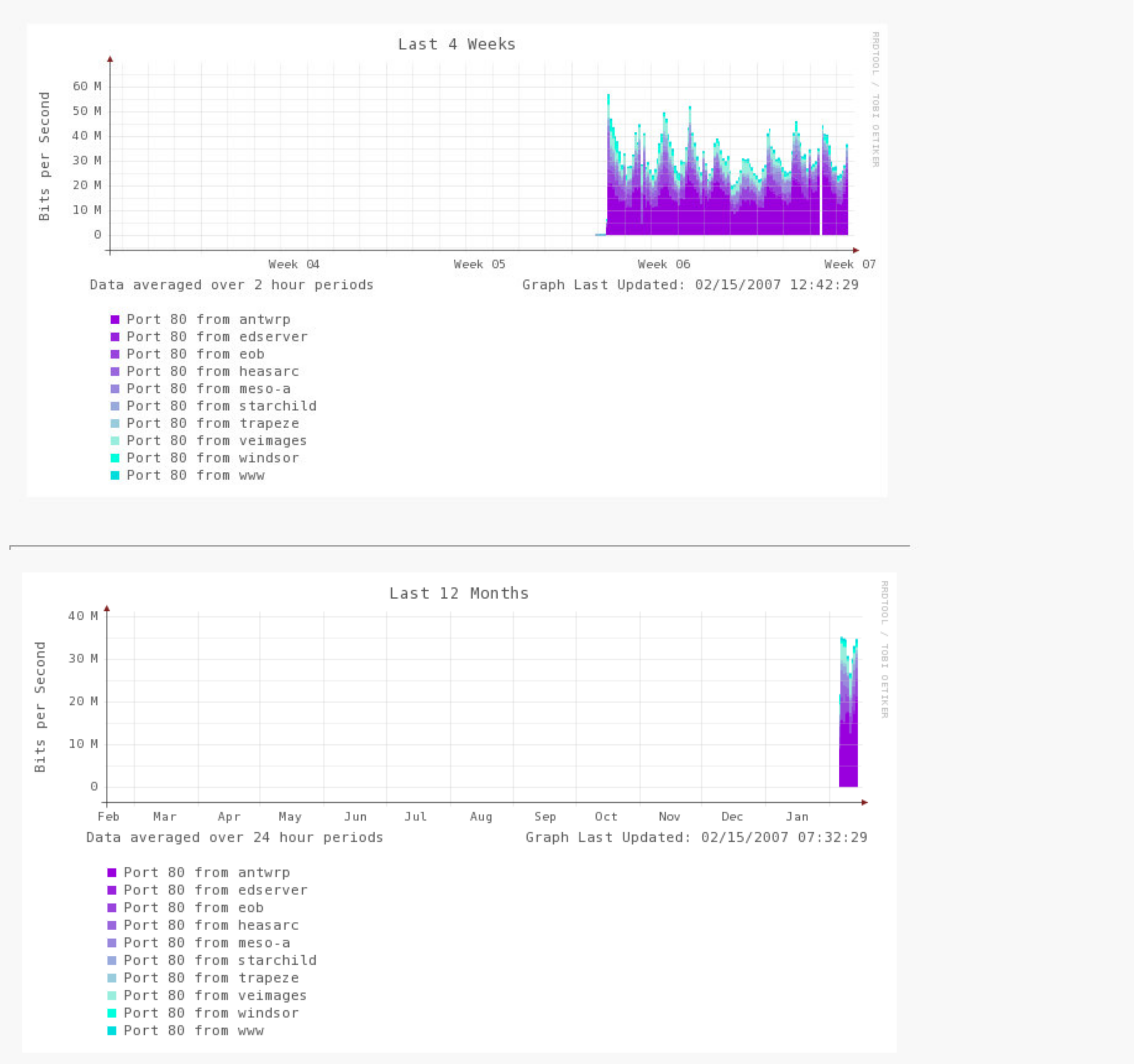


Figure 8-5 Typical FlowTracking Group web page

Running FlowTracker_Collector and FlowTracker_Grapher

After un-tarring the FlowViewer package and modifying the FlowViewer_Configuration.pm file for their environment, the user should initiate the FlowTracker_Collector and FlowTracker_Grapher programs from the command line. Each of these programs is intended to run continuously so they should be put into the 'background' (e.g. " somehost >FlowTracker_Collector&.") The user does not need to start these programs on any particular minute, as they will self-adjust to collect on even five minute intervals, and graph when first started and every graphing_period seconds thereafter.

Note that these processes will have to interact with files that have been established by the web server, so that the permissions on the web-server created files, (i.e., in particular the RRD files) must allow the user who starts the FlowTracker_Collector and FlowTracker_Grapher processes to be able to write to these files.

FlowTracker_Collector controls itself to run every five minutes. Once started, the script first looks to see if has been started less than five minutes from its previous execution. If so, it will go to sleep until a full five minutes has elapsed since its last execution. FlowTracker_Collector will then parse through each of the established Trackings (identified by the presence of a Tracking filter file in the FlowTracker_Filters subdirectory) reading the filter file and invoking:

```
flow-cat
flow-nfilter
flow-print (132 columns)
```

FlowTracker_Collector will reuse previously created flow-tools concatenation files to speed up the processing. This is accomplished on a device-by-device basis, that is, concatenation files will be generated for each device one time only.

The output file is parsed and flow data for the 5-minute period is accumulated. Parsing individual flows in this manner is necessary to accumulate the correct portion of flows that cross either or both ends of the 5-minute time period. The script then invokes rrdtool_update with this latest data point. The RRDtool file for each active tracking is stored in the FlowTracker_RRDtool subdirectory established by the user.

As of version 3.3, FlowTracker_Collector will compare the rate it collected for a tracking against eth Alert Threshold if this has been established. If a threshold has be set up and the value determined for the collection period exceeds it (or has not met it for negative threshold values), an email is sent to the email address established along with the Alert Threshold.

After FlowTracker_Collector has completed this process for each active Tracking, it determines how much time it took to do this and subtracts that from five minutes and puts itself to sleep for what remains of the period. If the user has elected to log FlowTracker_Collector activity, the script will output information to the FlowTracker_Collector.log file.

After starting up FlowTracker_Collector, the user should invoke FlowTracker_Grapher from the command line, placing it also in the ‘background’ (e.g., ‘somehost >FlowTracker_Grapher&’.)

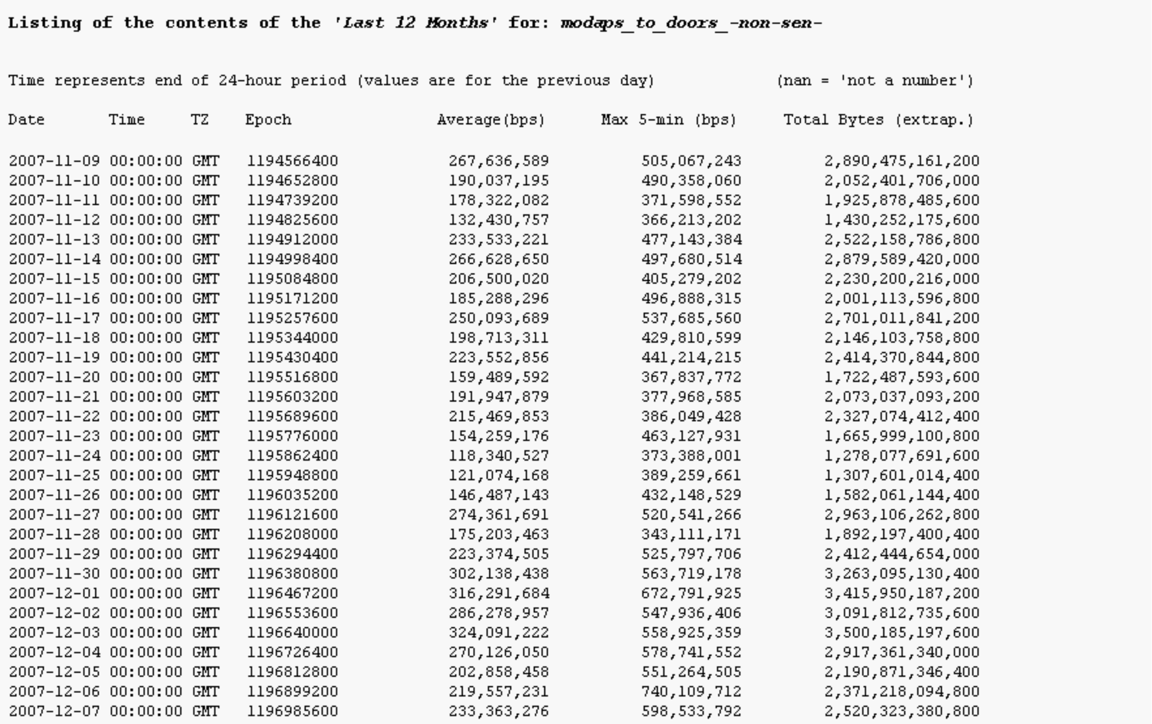
FlowTracker_Grapher simply invokes rrdtool_graph to create the four MRTG-like graphs for each active tracking and goes back to sleep for a parameter adjustable period (e.g., $graphing_period = 300;.) FlowTracker_Grapher updates only the MRTG-like graphs that have changed if the $lazy-mode variable is set.

Each time FlowTracker_Grapher runs it re-builds each individual tracking HTML file and the overall active trackings HTML page ($actives_webpage.) This page is provided as a single point of entry for other users to be able to link to each active Tracking. Note: the user may wish to create their own “overall” web page.

From the Tracking HTML page, the user can click on the FlowViewer and FlowGrapher portion of the FlowTracker image at the top. These will invoke either FlowViewer.cgi, or

FlowGrapher.cgi with filter criteria pre-filled with the Tracking filter criteria. This permits additional analysis.

As of version 3.3, user's can now generate a textual listing of the data making up each graph in the FlowTracking. Embedded in each graph is a [List Values] option. the resulting web page lists the values and also does an approximation (extrapolation: bits/seconds * seconds) of Bytes transferred during the period. An example from a "Yearly" graph showing bytes per day is shown in figure 8-7 below.



```
Listing of the contents of the 'Last 12 Months' for: modaps_to_doors_-non-sen-

Time represents end of 24-hour period (values are for the previous day)            (nan = 'not a number')

Date       Time     TZ    Epoch               Average(bps)      Max 5-min (bps)     Total Bytes (extrap.)

2007-11-09 00:00:00 GMT   1194566400           267,636,589         505,067,243         2,890,475,161,200
2007-11-10 00:00:00 GMT   1194652800           190,037,195         490,358,060         2,052,401,706,000
2007-11-11 00:00:00 GMT   1194739200           178,322,082         371,598,552         1,925,878,485,600
2007-11-12 00:00:00 GMT   1194825600           132,430,757         366,213,202         1,430,252,175,600
2007-11-13 00:00:00 GMT   1194912000           233,533,221         477,143,384         2,522,158,786,800
2007-11-14 00:00:00 GMT   1194998400           266,628,650         497,680,514         2,879,589,420,000
2007-11-15 00:00:00 GMT   1195084800           206,500,020         405,279,202         2,230,200,216,000
2007-11-16 00:00:00 GMT   1195171200           185,288,296         496,888,315         2,001,113,596,800
2007-11-17 00:00:00 GMT   1195257600           250,093,689         537,685,560         2,701,011,841,200
2007-11-18 00:00:00 GMT   1195344000           198,713,311         429,810,599         2,146,103,758,800
2007-11-19 00:00:00 GMT   1195430400           223,552,856         441,214,215         2,414,370,844,800
2007-11-20 00:00:00 GMT   1195516800           159,489,592         367,837,772         1,722,487,593,600
2007-11-21 00:00:00 GMT   1195603200           191,947,879         377,968,585         2,073,037,093,200
2007-11-22 00:00:00 GMT   1195689600           215,469,853         386,049,428         2,327,074,412,400
2007-11-23 00:00:00 GMT   1195776000           154,259,176         463,127,931         1,665,999,100,800
2007-11-24 00:00:00 GMT   1195862400           118,340,527         373,388,001         1,278,077,691,600
2007-11-25 00:00:00 GMT   1195948800           121,074,168         389,259,661         1,307,601,014,400
2007-11-26 00:00:00 GMT   1196035200           146,487,143         432,148,529         1,582,061,144,400
2007-11-27 00:00:00 GMT   1196121600           274,361,691         520,541,266         2,963,106,262,800
2007-11-28 00:00:00 GMT   1196208000           175,203,463         343,111,171         1,892,197,400,400
2007-11-29 00:00:00 GMT   1196294400           223,374,505         525,797,706         2,412,444,654,000
2007-11-30 00:00:00 GMT   1196380800           302,138,438         563,719,178         3,263,095,130,400
2007-12-01 00:00:00 GMT   1196467200           316,291,684         672,791,925         3,415,950,187,200
2007-12-02 00:00:00 GMT   1196553600           286,278,957         547,936,406         3,091,812,735,600
2007-12-03 00:00:00 GMT   1196640000           324,091,222         558,925,359         3,500,185,197,600
2007-12-04 00:00:00 GMT   1196726400           270,126,050         578,741,552         2,917,361,340,000
2007-12-05 00:00:00 GMT   1196812800           202,858,458         551,264,505         2,190,871,346,400
2007-12-06 00:00:00 GMT   1196899200           219,557,231         740,109,712         2,371,218,094,800
2007-12-07 00:00:00 GMT   1196985600           233,363,276         598,533,792         2,520,323,380,800
```

Figure 8-7 List Values option from a "Yearly" FlowTracker graph

9. *Cleaning Up*

The following files are provided in the distribution for cleaning up caches and directories of Reports and Graphs that have lost their usefulness:

- □ *FlowViewer_CleanASCache*

This script is used to remove Autonomous System resolutions that may have changed externally, but remain in the FlowViewer AS Cache file. It is invoked from the command line.

- □ *FlowViewer_CleanHostCache*

This script is used to remove host name resolutions that may have changed in DNS, but remain in the FlowViewer Names Cache file. It is invoked from the command line.

□ *FlowViewer_CleanFiles*

This script is used to clean up older files that remain in the Reports, Graphs, and Trackings directories. When a user saves a report or graph in version 3.3, it is saved to a separate directory, so the Reports and graphs directories can now be cleaned without fear of removing saved reports or graphs. This file can be invoked daily from crontab. An example is shown below:

```
#  Setting up crontab file for cleaning FlowViewer files:
#
#   min   hr  dom  moy  dow    command
#
#     5   0    *    *    *     /htp/cgi-bin/FlowViewer_3.3/FlowViewer_CleanFiles
#                                > /htp/cgi-bin/FlowViewer_3.3/cleanup.log
#                                2 >> /htp/cgi-bin/FlowViewer_3.3/cleanup.log
```